INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA INFORMÁTICA




Decision Support System for facility location problems in fleet management

JOÃO MIGUEL FARIA MARTINS

Outubro de 2020

POLITÉCNICO
DO PORTO

Decision Support System for facility location problems in fleet management

João Miguel Faria Martins

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Computer Systems

Supervisor: Prof.ª Dra. Goreti Marreiros
Co-Supervisor: Prof. Dr. Carlos Ferreira

Porto, October 14, 2020

Dedicatory

Dedicated to everyone that I truly know.

Abstract

Businesses that are growing by providing more services, reaching more customers or improving their business strategy, might need to create or relocate a facility location to expand the geographical coverage and improve their services. This decision is complex, and it is crucial to analyse their client locations, journeys and be aware of the factors that may affect their geographical decision. These organisations must weigh all these factors, such as security levels, taxes or costs due to the importance and impact that they can have in the short and long term business strategy. Therefore, the decision-maker needs to ensure that the location is the most profitable site according to the business scope and future perspectives.

To help the businesses on this complex decision, this dissertation details the development of a Decision Support System (DSS) capable of providing facility location suggestions based on the existing journeys and the factors that the decision-maker considers more relevant to the company.

The developed DSS has three main components: (1) Decision Support System; (2) Geospatial Analysis System; and (3) Facility Location Factors System. The Decision Support System is responsible for producing ordered facility location suggestions by performing the multi-criteria decision analysis (MCDA) that implements the TOPSIS algorithm. The Geospatial Analysis System, through the use of the DBSCAN algorithm, is responsible for retrieving the alternatives by identifying the geospatial clusters based on the existing journeys. Lastly, the Facility Location Factors System is responsible for retrieving the criterion by gathering the data from external sources according to the chosen factors.

The evaluation analysis shows that the perspective of the users about the assistance of the system by helping them choose appropriate facility location is favourable. This analysis showed that the users agree about the accuracy and the value of the facility location suggestions.

The output helps the business managers to make better decisions by returning facility locations that have potential to maximise the company's profit by reducing transportation and fuel costs and maximise the number of covered customers by expanding their territorial coverage.

This project handles data provided by Fonix Telematics from their United Kingdom clients that have relevance to the study, such as a high number of assets, journeys and geographical coverage.

Keywords: Fleet Management, Decision Support System, Geospatial Analysis, Facility Location Problem

Resumo

As empresas em crescimento por via da disponibilização de mais serviços, do aumento do seu leque de clientes ou da melhoria da sua estratégia, podem pretender criar ou realocar um centro de operações de modo a expandir a sua cobertura geográfica e, consequentemente, melhorar os seus serviços. Esta decisão é complexa e é fundamental analisar vários aspetos, assim como, a localização dos seus clientes, as viagens recorrentes e, acima de tudo, estar consciente dos fatores que podem afetar a sua decisão geográfica. As organizações devem pesar todos esses fatores, assim como níveis de segurança, impostos ou custos, devido à importância e ao impacto que podem ter na estratégia da empresa a curto e a longo prazo. Portanto, o decisor necessita de garantir que o local é rentável e que capta o âmbito do negócio e as perspetivas futuras.

De modo a auxiliar as empresas nesta complexa decisão, esta dissertação detalha o processo de desenvolvimento de um Sistema de Apoio à Decisão (SAD) capaz de fornecer um conjunto de sugestões com os locais mais indicados para a criação de um centro de operações com base nas viagens efetuadas e nos fatores que o decisor considera mais relevantes para a organização.

O SAD desenvolvido possui três componentes: (1) *Decision Support System*; (2) *Geospatial Analysis System*; e (3) *Facility Location Factors System*. O *Decision Support System* é responsável por produzir as sugestões geoespaciais, através da Análise de Decisão Multicritério (MCDA) que por sua vez implementa o algoritmo TOPSIS. O *Geospatial Analysis System*, através da utilização do algoritmo DBSCAN, é responsável por retornar as alternativas através da identificação dos *clusters* geográficos com base nas viagens existentes. Por último, o *Facility Location Factors System* é responsável por retornar os critérios, que são compostos por dados recolhidos através de fontes externas de acordo com os fatores previamente selecionados.

A avaliação da solução demonstra que a perspetiva dos utilizadores sobre o sistema é positiva e que, de facto, os auxilia na decisão do local mais indicado para as suas instalações. A análise indica ainda que os utilizadores estão de acordo com a precisão e com locais sugeridos para os centros de operações.

Estas sugestões auxiliam os decisores a tomarem decisões mais sustentadas, visto que os locais sugeridos possuem potencial para maximizar a rentabilidade da empresa, reduzir os custos de transporte e combustível, assim como maximizar a cobertura de clientes através do posicionamento geográfico.

Este trabalho utiliza dados de clientes da Fonix Telematics que atuam no Reino Unido e que possuem relevância para o estudo, como um número significativo de veículos, viagens e cobertura geográfica.

Acknowledgement

I would first like to thank my supervisors, Professor Goreti Marreiros and Professor Carlos Ferreira, for their valuable guidance throughout my thesis. You provided me with the tools that I needed to choose the right direction and successfully completed my dissertation. Your feedback brought my work to a higher level.

Also, I would like to thank my supervisor from Fonix Telematics, Eng. Gonçalo Oliveira, for his guidance through each stage of the process. I would also like to thank him for the shared expertise during the development process.

Finally, I would like to express my acknowledge to my family and friends for being always there for me. Particularly to my friend, Xavier Silva, who provided stimulating discussions to this dissertation.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AHP	**A**nalytical **H**ierarchy **P**rocess
AI	**A**rtificial **I**ntelligence
CAGR	**C**ompound **A**nnual **G**rowth **R**ate
DI	**D**ependency **I**njection
DSS	**D**ecision **S**upport **S**ytem
FLP	**F**acility **L**ocation **P**roblem
FMS	**F**leet **M**anagement **S**ystem
IoC	**I**nversion **o**f **C**ontrol
MCDA	**M**ulti **C**riteria **D**ecision **A**nalysis
MCDM	**M**ulti **C**riteria **D**ecision **M**aking
ML	**M**achine **L**earning
NCD	**N**ew **C**oncept **D**evelopment
SAM	**S**erviceable **A**vailable **M**arket
SOM	**S**erviceable **O**btainable **M**arket
TAM	**T**otal **A**vailable **M**arket
TOPSIS	**T**echnique for **O**rder of **P**reference by **S**imilarity to **I**deal **S**olution

Chapter 1

Introduction

This chapter has a description of the context, a brief description of the problem, the objectives and the approach for the development of this project.

1.1 Context

A fleet management solution allows the clients to track their vehicles at any time with the information being transmitted back in real-time through cellular networks that offer reliable broadband communication (Rantell 2019).

Getting the best from a fleet and improving operational and financial performance is imperative. Data is one of the most important aspects of any fleet business. It helps to drive companies forward, keep the business flowing and optimise efficiencies. Collecting, interpreting and acting on data helps the businesses make better decisions. The collected data can demonstrate how their business is performing, allowing them to decide on how to improve. Understanding data has allowed companies to turn raw data into trends for their business and are even able to turn that data into predictions for the future too (Rantell 2019).

The purpose of this project is to increase the value of Fleet Management solutions by adding an intelligent mechanism that can help companies to make better decisions based on the telemetry coming from their assets.

When a company is interested in creating or relocating a facility location, need to be aware of their business scope and all the related restrictions. It is imperative to analyse their client locations, journeys and be aware of the factors that may affect their geographical decision. These organisations must weigh all these factors (such as security levels, taxes or costs) due to the importance and impact that they can have in the long term business strategy.

Thus, this project can help decision-makers by sustaining their decisions through the use of a mechanism that is capable of generating facility location suggestions.

The selection of the facility location is crucial because the decision-maker needs to ensure that the location is the most profitable site according to the business scope and long term strategy. The location should satisfy the business needs and cover the maximum number of customers to provide better services with existing clients and even to expand their services to the new location (Kant et al. 2018).

1.2 Problem

Facility location-allocation is a critical element and plays a role in the strategic design of the business company's network (Melo, Nickel, and Saldanha-da-Gama 2009). A Facility Location Problem (FLP) involves a set of spatially distributed customers and a set of facilities to serve customer demands, and it tries to define the most valuable location for the business to cover customer demand.

High costs associated with property acquisition and facility construction make the facility location or relocation projects a long-term investment. Decision-makers must select sites that will not merely perform well according to the current system state, but that will continue to be profitable for the facility's lifetime, even considering environmental factors change, populations shift, and market trends evolve (Owen and Daskin 1998).

Fleet companies that are growing might need to create or relocate their facility locations to expand the geographical coverage and their services. These businesses must evaluate which sites are the most valuable for them. This decision of creating or relocating a facility location is complicated and time-consuming. The company must know their most journeys, their customer locations and the factors that may affect the business interests. The complexity of understanding these factors such as taxes, construction or rent costs per m^2, security levels, growth zones and fuel costs, fulfil the importance of a mechanism that is capable of making a multi-criteria analysis.

It is clear to recognise that the selection of the facility location is a complex process. The decision made exclusively by the decision-makers knowledge can be time-consuming and unreliable due to the number of constraints and its complexity.

This project uses data provided by Fonix Telematics from their United Kingdom clients that have relevance to the study, such as a high number of assets, journeys and geographical coverage. Therefore, this dissertation uses data from the two most relevant companies named as Company A and Company B. Company A has a total of 161.220 journeys and 8.168.429 positions, and Company B has a total of 140.535 journeys and 5.964.332 positions.

1.3 Objectives

The primary goal of this project is to support and help a fleet business in the decision of the location of an operations facility. This project helps these companies by proving a list of facility location suggestions built according to the business scope and needs. A Decision Support System (DSS) can help these decision-makers to accomplish better decisions and potentially increase fleet performance, productivity and reduce operating expenses based on their fleet data.

The main objectives of this work to achieve the project purposes are:

- Study the problem's state of the art;
    - Facility location problems;
    - Machine Learning - focusing on unsupervised learning;
    - Decision Support System - focusing on multi-criteria decision analysis (MCDA) techniques;
    - Technologies - focusing on clustering algorithms;

- Design and conception of a Decision Support System;
  - Implement a system of geospatial clustering analysis capable of identifying potential facility locations;
  - Study the factors that can affect the long-term facility location decision. Implement the identified and most relevant factors to be used by the DSS.
  - Implement a MCDA technique to produce ordered facility location suggestions.
- Construction and development of a Decision Support System;
- System integration with real telemetry data being transmitted by the assets;
- Evaluate and test the solution. The evaluation focus on the quality analysis and performance of the DSS, and the metrics of the clustering algorithm.

The output of the solution should help the business managers to make better decisions by returning locations that have potential to maximise the company's profit and the number of covered customers by expanding their territorial coverage and minimise some company's cost.

1.4 Approach

This dissertation focuses on the development of a Decision Support System (DSS) to produce valuable suggestions to help the decision-makers in a facility location problem. The objective includes that the density of destination points in a radius and the facility point is the most suitable point of the dense cluster which optimise the transportation and infrastructure costs. The solution currently uses data provided by Fonix Telematics and is capable of producing suggestions to various companies with distinct needs and scopes.

The objectives of this work were analysed and gone through a rigorous design process for the further development process. The DSS has three main components:

1. Generation of alternatives based on geospatial clustering analysis;
2. Criterion selection based on the analysis of the factors that may affect the facility location;
3. Ordering facility location suggestions as a result of the multi-criteria decision analysis (MCDA).

First, the generation of alternatives is relative to the identification of geospatial clusters and is achieved by analysing the data coming from the assets. This data should be processed to ensure its robustness and suitability for the use of intelligent data analysis techniques such as clustering techniques. The geospatial suggestions lead to a facility location problem and consequently to the study of the factors that affect that choice.

According to the current state of the art, the Facility Location Problem (FLP) was already extensively studied and represent a topic with high relevance in the investigation field. Most of them lead to an NP-Hard problem due to the complexity of the subject. The FLP literature demonstrates different approaches to provide an optimal solution, and each approach has corresponding factors that affect the facility location-allocation process.

Second, the factors are relative to security and taxes levels or construction costs. The decision-makers choose the factors that are more relevant for them according to their business context. These criteria may vary from customer to customer, and it is supposed to allow them to select the most relevant ones based on a predefined list of supported factors. These supported factors were studied and represent the result of an intensive study made by Fonix Telematics experts.

Third, based on the produced geospatial clusters and the most relevant factors that can influence the location, the DSS can provide ordered facility location suggestions that can support the business manager to make a sustained decision and guide the company through a decision-making process. The solution is prepared to support data from any company that is using a fleet management system. The suitability and the system's quality is verified by testing against various scenarios.

The system evaluation focuses on the quality and performance analysis of the DSS, and the metrics of the clustering algorithm. In order to accomplish these goals, the solution is tested in different scenarios to make sure the customer needs are fulfilled.

1.5 Document Structure

The document is divided into the following chapters: 1. Introduction; 2. State of the Art; 3. Value Analysis; 4. Design; 5. Implementation; 6. Evaluation; 7. Conclusion.

The Introduction has a description of the context, a brief description of the problem, the objectives and the approach for the development of this project.

The State of the Art concerns the function of describing the highest level of development or scientific field at a particular time. Also has comparative studies for the most relevant subjects of this project. This section is composed by the: problem description and context; facility location problems; machine learning overview focusing on unsupervised learning by describing and comparing clustering techniques; description of DSS types and techniques - description and comparison of Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and Analytical Hierarchy Process (AHP) methods; study of the existing technologies and a comparative study to identify the most appropriate technologies to be used on the geospatial analysis system.

The Value Analysis concerns the function of a product to meet the demands or application need by a customer. The value was analysed based on the: Innovation Process using the New concept development (NCD) model by P. Koen et al. 2001; calculation of the TAM, SAM and SOM; value of the solution for the customer and the perceived value; definition of the value proposition; Business Canvas Model; Quality Function Deployment (QFD). This section explores those components in detail.

The Design details the following items: requirement analysis divided into functional requirements and non-functional requirements; system architecture with the proposed components and deployment views of the system; architecture and technologies for each subsystem; main flow execution of the solution.

The Implementation concerns the function of describing and discussing the implementation of the solution according to the system analysis and design detailed in the Design chapter following good practises of software development.

The Evaluation concerns the function of describing the process of evaluation and analysis of the solution. The first step is regarding the hypothesis identification, then a description of the evaluation indicators, the explanation of the evaluation methodology and, finally, the evaluation of experiments and results.

The Conclusion concerns the function of concluding this dissertation by giving an overview of the developed work, detailing the achieved requirements, discussing the limitations, enumerating and describing the future work and a final appreciation.

Chapter 2

State of the Art

The State of the Art concerns the function of describing the highest level of development or scientific field at a particular time. Also has comparative studies for the most relevant subjects of this project.

This chapter is composed by the: problem description and context; facility location problems; machine learning overview focusing on unsupervised learning by describing and comparing clustering techniques; description of Decision Support System (DSS) types and techniques - description and comparison of Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and Analytical Hierarchy Process (AHP) methods; study of the existing technologies and a comparative study to identify the most appropriate technologies to be used on the geospatial analysis system.

2.1 Problem

Fleet management is the process that fleet managers use to manage all fleet and asset data. Getting the best from a fleet and improving operational and financial performance is imperative. Data is one of the most important aspects of any fleet business. It helps to drive companies forward, keep the business flowing and optimise efficiencies. Collecting, interpreting and acting on data helps the clients make better decisions. The collected data can demonstrate how their business is performing, allowing them to decide on how to improve. Understanding data has allowed companies to turn raw data into trends for their business and are even able to turn that data into predictions for the future too (Rantell 2019).

When fleet companies are growing, and they need to create or relocate their facility locations in order to expand the geographical coverage and their services, they need to analyse their client locations, their common journeys and be aware of the factors that may affect their geographic decision. These organisations must weigh these factors, such as council tax indexes, indices of deprivation, construction or rent costs, house price indexes, security levels or crime rating, when choosing a facility location because they may affect the long-term business strategy.

Facility location-allocation is a critical element and plays a role in the strategic design of business company's network (Melo, Nickel, and Saldanha-da-Gama 2009). Facility Location decisions are a critical element in strategic planning for many companies and often challenged by difficult spatial resource allocation decisions. Siting facilities are based and long-lasting, impacting numerous operational and logistical decisions. Furthermore, property acquisition and facility construction make facility location or relocation projects long-term investments

(Owen and Daskin 1998). Decision-makers must select sites that will not merely perform well according to the current system state, but that will continue to be profitable for the facility's lifetime, even as environmental factors change, populations shift, and market trends evolve (Owen and Daskin 1998). Analysing all these constraints with a high level of complexity is often difficult without the support of any intelligent system capable of providing suggestions to support their decisions.

Providing an intelligent system capable of providing ordered facility location suggestions can reduce the risk and increase the chances of making the right decision. The most suitable and strategic choice can ease the maximisation of the profit by reducing transportation and fuel costs and maximising the number of covered customers and potentially expand their territorial coverage. By making use of existing telemetry data, the value of Fleet Management solutions can be increased, and support companies on their complex decisions.

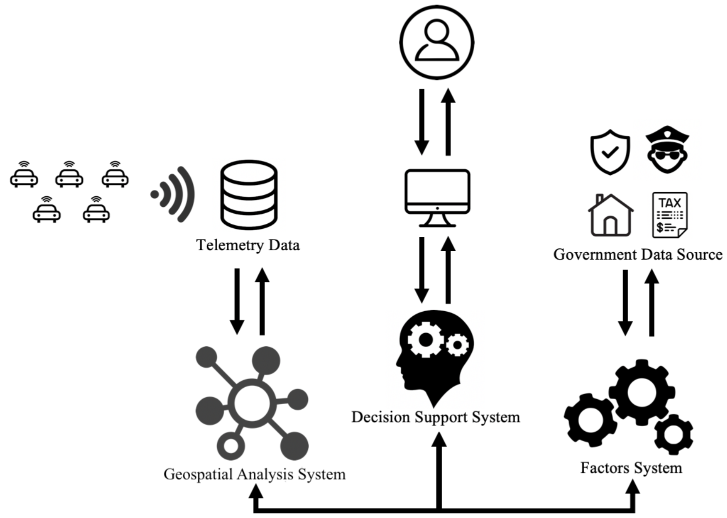


Figure 2.1: High Level System Architecture Representation

To simplify the understanding of the problem and the components of the solution, it was developed the diagram presented in Figure 2.1. Three main components compose the solution:

- Geospatial Analysis System is responsible to process telemetry data and to identify geospatial clusters. This data is sent to the Decision Support System to be used as alternatives in the multi-criteria decision analysis;
- Facility Location Factors System is responsible for getting data from government data sources to each identified cluster. This data is sent to the Decision Support System to be used as criterion in the multi-criteria decision analysis;
- Decision Support System is responsible for producing facility location suggestions through the implementation of the multi-criteria decision analysis based on the decision-maker request.

To analyse journeys and identify locations with the potential to become a facility location, it is fundamental to analyse telemetry data at least from the last six months to produce cohesive suggestions. This document's work is based on data provided by Fonix Telematics from their United Kingdom clients that have relevance to this dissertation, such as companies with a high number of assets, journeys and geographical coverage.

2.1.1 Telemetry Data Structure

The telemetry data used by the solution to identify the geospatial clusters that can become a facility location is divided into two separate data structures: (1) journeys and (2) telemetry data.

The table 2.1 details the journeys data structure. This structure is regarding the summary of the completed journeys and, with this data, it is possible to identify starting and ending locations, total mileage, fuel cost and usage.

Table 2.1: Simplified journeys data structure

Key	**Description**
AssetId	Asset Unique Identification
AccountId	Company Unique Identification
DataPoints	Total number of positions
EndGpsTimestamp	End time of journey
EndAddress	End journey address
FuelUsed	Fuel used in the journey
FuelCost	Fuel cost of the journey
Mileage	Journey mileage (km)
StartGpsTimestamp	Start time of journey
StartAddress	Start journey address

The telemetry data structure is detailed in the table 2.2 and its regarding to each transmitted position. Thus, one journey has n positions.

Table 2.2: Simplified telemetry data structure

Key	**Description**
AssetId	Asset Unique Identification
AccountId	Company Unique Identification
GpsTimestamp	Time of the transmitted data
Ignition	Ignition status
Latitude	Latitude
Longitude	Longitude
Speed	Speed

2.1.2 Decision Making Factors

Facility location must be appropriately selected by decision-makers while planning to set up their business. This decision is a long term decision with high impact in the business strategy.

Table 2.3: Factors for facility location selection

Factor	Details/Metrics	Criteria Type
Construction Costs	Material prices, price per m^2 and wages	Cost
Council Tax Band	Value evaluation of the property	Cost
Fuel costs	Evolution of the fuel cost	Cost
Growth zones	Location evolution and investment	Benefit
House Price Index	Price changes and average prices	Cost
Indices of Deprivation	Employment Rank, Education and Skills, Health and Living environment	Benefit and/or Cost
Proximity to customers	Distance of the facility location to potential customers	Benefit
Safety levels	Crime rank per area	Cost

The decision-maker must know the factors that affect the business and are more relevant to them.

The factors are classified as controllable/benefit (meaning more is better) and uncontrollable/cost (meaning less is better). The factors were analysed by Fonix Telematics experts and this dissertation focus on these factors (c.f. table 2.3).

2.2 Facility Location Problems

Facility Location decisions are a critical element in strategic planning for many companies and are often challenged by difficult spatial resource allocation decisions. Siting facilities are based on long-lasting, impacting numerous operational and logistical decisions. Furthermore, property acquisition and facility construction make facility location or relocation projects long-term investments (Owen and Daskin 1998).

Facility location problems locate a set of facilities to minimize the cost of a set of demands taking consideration of some constraints (Farahani and Hekmatfar 2009). The decision-makers must select sites that will perform well for the facility's lifetime. Finding a profitable location is a complex task because decision-makers need to be aware of uncertain future events. Four components describe location problems: customers, facilities, space in which customer and facilities are located, and a metric defining the distance or time between them (Farahani and Hekmatfar 2009).

Three categories address facility location problems: (1) static and deterministic models (inputs are taken as known quantities and outputs are specified as one-time decision values); (2) stochastic (tries to capture the complexity and uncertainty of real-world problems); or (3) dynamic (focus on the complicated timing problems involved in locating facilities over an extended time) models (Owen and Daskin 1998). Due to the complexity of the problem, researchers develop over the years, a vast number of mathematical models to represent a wide variety of location problems (Farahani and Hekmatfar 2009). The discussion in this section about Facility Location Problems has been influenced by the book written by Farahani and Hekmatfar - more details, and other facility location methods can be found in the book (Farahani and Hekmatfar 2009).

The resulting models can be difficult to solve to optimality[1] resulting in NP-hard problems (Owen and Daskin 1998). This project focus only on the models with the potential to respond to the problem. The first model is the single facility location problem introduced by Alfred Weber, considered to be the foundation of modern location theories. The second model is the covering problem, which the customer needs to be within a specified radius. Finally, are described the centre and the location-allocation problems.

2.2.1 Single Facility Location Problem

These problems occur very often when choosing the location of a warehouse to serve customers. This approach should be used when the company has limited resources for decision analysis. The considered objective function in this scenario is the total travel distance or total travel cost (Farahani and Hekmatfar 2009).

To calculate the objective function is used the Euclidean distance to measure the distance between the new facility and a collection of customers (Farahani and Hekmatfar 2009). The use of these formulas is not precise in geospatial scenarios because they calculate the segment between two points with a straight-line distance. To measure distances between coordinates and get the great-circle distance[2] it should be used the Haversine Formula.

2.2.2 Covering Problem

In covering problems, the company's service depends on the distance between the facility and the customer. The facility only provides their service if the distance between the customer and the facility is equal or less than a predefined distance. The critical value is designated coverage distance or coverage radius (Farahani and Hekmatfar 2009).

This problem has two branches known as tree networks, and general networks and each of these branches has two problems based on the covering demand points: total covering (covers all points) and partial covering (covers some points) problems. (Farahani and Hekmatfar 2009).

2.2.3 Center Problem

In the covering problems, the objective is to determine the location of the minimum number of facilities to cover all demand points. One problem related to this is the fact that the number of facilities can be significant. In this model, the goal is to maximize the number of covered demand nodes with a fixed number of facilities. The objective is to find the location of *n* facilities to cover all the demand points and minimize the distance between a demand point and the nearest facility (Farahani and Hekmatfar 2009).

2.2.4 Location Allocation Problem

Location allocation problems desire to locate a set of new facilities to reduce transportation costs from facilities to customers, and an optimal number of facilities have to be placed to satisfy the customer demand. The main components of this problem are facilities, locations and customers. The facilities are usually characterized by their capacity, profit, type of service and costs (Farahani and Hekmatfar 2009).

[1] noun: best or most favourable; optimum: seeking the optimal solution.

[2] Is the shortest distance between two points on the surface of a sphere

2.3 Machine Learning

Machine Learning (ML) is an Artificial Intelligence (AI) branch, that focuses on developing algorithms to teach how computers learn from data to make decisions or predictions. These procedures operate by the construction of a model from inputs to make data-driven predictions rather than following static program instructions (Simon et al. 2015).

This term refers to the automated detection of meaningful patterns in data. ML is applied in various scenarios: search engines that find the best results, anti-spam software that learns to filter our email messages, cars equipped with accident prevention systems and digital cameras to detect faces (Salkind 2013).

If a given problem is complex and the tasks are too complex to be performed by humans and sometimes beyond human capabilities, the ML can be used when the use of programs that learn and improve based on their "experience" can resolve the problem. The adaptivity[3] is another reason to use ML solutions because one limiting feature of programmed tools is their rigidity and many tasks change over time (Salkind 2013).

ML has three main branches of learning methods: supervised (c.f. section 2.3.1), unsupervised (c.f. section 2.3.3) and reinforcement learning (c.f. section 2.3.2). This document has a brief description of supervised and reinforcement learning and describes in detail the unsupervised learning focusing on clustering techniques.

2.3.1 Supervised Learning

Supervised learning is the computational task of learning by making correlations between variables present in the training set and using this information to create a predictive model capable of inferring new data (Fabris, Magalhães, and Freitas 2017).

The defining characteristic of supervised learning is the availability of annotated training data. The supervisor instructs the learning system on the labels to associate with training examples. Typically these labels are class labels in classification problems. Supervised learning algorithms induce models from these training data, and these models can be used to classify other unlabelled data (Cunningham, Cord, and Delany 2008).

There are two main categories of supervised learning:

- Classification: Predict discrete values (e.g. predicting the marital status of a person);
- Regression: Predict continuous values (e.g. home prices).

In summary, this technique entails learning a mapping between a set of input variables and an output variable and applying this mapping to predict the outputs for unseen data (Cunningham, Cord, and Delany 2008).

2.3.2 Reinforcement Learning

Reinforcement learning (RL) is an area of ML concerned with how software agents ought to take actions in an environment in order to maximize some notion of cumulative reward. RL techniques have three key features: the problems need to understand what to do, how to map situations to actions, and how to maximize a numerical reward signal (Sutton and Barto 2015). Figure 2.2 shows the typical flow execution of reinforcement learning.

[3] noun: characterized by or given to adaptation

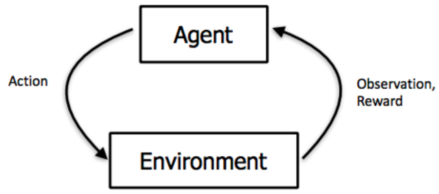


Figure 2.2: Reinforcement Learning flow execution (Mayank 2018)

In summary, they are closed-loop problems because the learning system's actions influence their later inputs. Moreover, the learner do not know which actions to take, as in many forms of machine learning, but instead must discover which actions yield the most reward by trying them out. These three characteristics - being closed-loop, not having direct instructions as to what actions to take, and where the consequences of actions, including reward signals are the three most important distinguishing features of reinforcement learning problems (Sutton and Barto 2015).

2.3.3 Unsupervised Learning

Unsupervised learning uses procedures that attempt to find partitions of patterns (Nilsson 1998). This technique tries to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution (Dönmez 2013).

The most used approach is the clustering technique that uses classes, or conceptually meaningful groups of objects that share common characteristics, play an essential role in how people analyze and describe the world. Indeed, human beings are skilled at dividing objects into groups (clustering) and assigning particular objects to these groups (classification) (Salkind 2013).

The utility of clustering is the use of cluster analysis to provide an abstraction from single data objects to clusters in which those data objects reside. Some clustering techniques use cluster prototype to define each cluster. Cluster prototype consists of the selection of a single element from the data space that represents a group of elements. So, cluster analysis wants to find the most representative cluster prototypes.

- Summarization: Apply an algorithm to a reduced data set with only cluster prototypes. Depending on the analysis, the results can be comparable to the results if all data points have been used (Salkind 2013).
- Compression: Cluster prototypes can be used to data compression. Create a table for each cluster with the assigned prototypes. Each prototype is assigned an integer that is its position in the table. This type is known as vector quantization and is often to image, sound and video data (Salkind 2013).
- Efficiently Finding Nearest Neighbors: Finding nearest neighbours can require the processing of the pairwise distance between all points. This search can be more efficient using the cluster prototypes. If objects are relatively close to the prototype of their cluster, using the prototypes can reduce the number of computations that are necessary to find the neighbours of an object (Salkind 2013).

In section 2.3.4, is described with more detail the main concepts of Clustering, various approaches to divide data into clusters, different types of clustering and clusters, and the essential clustering techniques.

2.3.4 Clustering

Cluster analysis divides data into clusters that are meaningful or useful. If meaningful groups are the goal, then the clusters should capture the natural structure of the data. However, cluster analysis can be useful to accomplish other purposes. Cluster analysis has long played a vital role in various fields: machine learning, pattern recognition, information retrieval, and data mining (Salkind 2013).

Clustering can be regarded as a form of classification, and it creates labeling of objects with class labels. It derives labels from the data and, for this reason, is referred to as unsupervised classification. In the next subsections, are detailed different approaches to clustering techniques. The main two categories are hierarchical clustering and partitional clustering. The partitional clustering focuses on centroid-based clustering and density-based clustering.

Hierarchical Clustering

In this method, clusters are composed by iterative applying of dividing patterns using top-down or bottom-up approach (Saxena et al. 2017). This algorithm applies this pattern to the data objects based on their distance to each other (Kumar and Verma 2018).

This algorithm has two forms of implementation, namely agglomerative and divisive hierarchical clustering (c.f. figure 2.3). The advantages of hierarchical clustering include acceptance of any forms of similarity or distance and apply to any attribute types. The main disadvantage of most of the hierarchical algorithms is that it does not revisit the built clusters to improve them (Berkhin 2013).

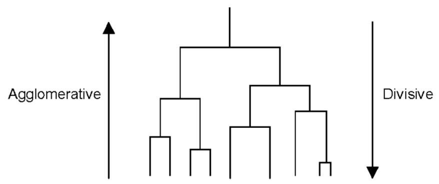


Figure 2.3: Agglomerative vs Divisive Hierarchical Clustering

The agglomerative method (c.f. algorithm 2.1) applies the bottom-up approach meaning this method starts with a single object and then merging these atomic clusters until all the objects are lying in a single cluster or until certain termination conditions are satisfied (Saxena et al. 2017).

Algorithm 2.1 Simplified agglomerative hierarchical clustering algorithm

```
repeat
    Merge closest two clusters
    Update the proximity matrix (this matrix is responsible to reflect the proximity/relation between clusters)
until One cluster remains
```

The divisive hierarchical method applies the top-down approach, which starts with a cluster containing all objects into smaller clusters, until each object forms a cluster or until certain termination conditions are satisfied (Saxena et al. 2017).

The hierarchical clustering methods can be grouped into three categories as follows: (1) Single-linkage clustering, (2) Complete-linkage clustering, and (3) Average-linkage clustering (c.f. figure 2.4).

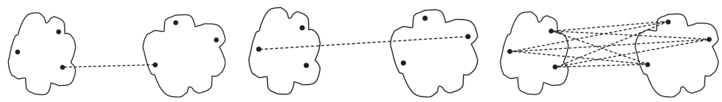


Figure 2.4: Hierarchical clustering categories by Salkind 2013

In the single-linkage clustering (c.f. figure 2.4.a), the link between two clusters is made by two elements that are closest to each other. The distance between two clusters is determined by the distance of these closest objects (Saxena et al. 2017).

In the complete-linkage clustering (c.f. figure 2.4.b), the distance between two clusters is determined by founding the longest distance of two objects in different clusters (Saxena et al. 2017).

In the average-linkage clustering (c.f. figure 2.4.c), the distance between two clusters is determined by the average pairwise proximity of all pairs of points from other clusters (Salkind 2013).

Partitional Clustering

The partitional clustering analyses data and separate objects into different partitions, known as clusters. This method is useful to represent large data sets and can improve data visualization comparing to tree structure used by hierarchical clustering (Saxena et al. 2017). The most used criteria to associate objects to clusters is the Euclidean distance, which finds the minimum distance between points and assigns the point to the available cluster (Saxena et al. 2017). The algorithms associated and studied in this type of clustering are the K-means (c.f. section 2.3.4), the DBSCAN and the OPTICS (c.f. section 2.3.4).

Centroid-based Clustering

Centroid-based clustering is a clustering method in which each cluster is a central vector, and the points are assigned to the clusters based on the proximity such that the squared distance from the central vector is minimized (Salkind 2013).

K-means Clustering

K-means algorithm defines a prototype (point) as a centroid typically applied to objects in a continuous n-dimensions space (Salkind 2013). This method divides the data into a user predefined number of clusters. The objective is to define one centroid for each cluster (Saxena et al. 2017).

The first step to implement a K-means algorithm (c.f. algorithm 2.2) is to define the K initial centroids, where K is the number of clusters and is defined by the user. In the algorithm execution, each point is assigned to the closest centroid, building a cluster. For each cluster, the centroid is updated based on the data points of that cluster. This process is repeated until the end of the points in the cluster, and the centroid was found (Salkind 2013). In the end, the natural grouping of points is identified and distributed by the predefined number of clusters.

Algorithm 2.2 Simplified K-means algorithm

1: Select K points as initial centroids
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid
4: Recompute the centroid of each cluster
5: **until** Centroids do not change

Time and Space Complexity

The time complexity of the K-means is $O(I * K * m * n)$, where I is the number of iterations made by the algorithm. The space complexity is $O((m + K)n)$ where m is the number of points and n is the number of properties. The data stored are only the centroids and the data points (Salkind 2013).

Strengths and Weaknesses

K-means clustering is the most used and explored clustering technique applied to a wide variety of data types and can be efficient. This technique cannot handle well arbitrary shapes or clusters with different sizes and densities. It isn't suitable for all types of data and has trouble handling data points that differs significantly from other observations - outliers. Finally, the notion of a centre is required, and it does not apply to all problems (Salkind 2013).

Density-Based Clustering

Density-based clustering locates regions of high density that are separated from each other by regions of low density. Objects form a dense region that is surrounded by a region of low density. This type of cluster is applied when the objects are irregular or intertwined, and the noise and outliers are present (Salkind 2013).

The traditional approach to analyse clusters density is the centre-based approach. In this approach, the density is estimated by counting the points within a specified radius, *Eps*, of the centred point. The density depends on the number of points included in the predefined radius (Salkind 2013).

The density centre-based approach classifies the points as being (1) a core point is in the interior of a dense region respecting the *Eps* radius and exceeds a user-specified threshold know as *MinPts*, (2) a border point is on the edge of a dense region, or (3) a noise point is an outlier to that density cluster.

DBSCAN Clustering

DBSCAN is a density-based algorithm (demonstrated by Ester et al. 1996) that produces a partitional clustering, discovers clusters with arbitrary shapes and is efficient for large spatial databases. This method does not produce a complete clustering[4], meaning the points in

[4] A complete clustering assigns every object to a cluster.

low-density are not associated with any cluster. These points are classified as noise and are omitted (Salkind 2013).

The DBSCAN algorithm (c.f. algorithm 2.3) can be described as follows: any two core points that are close enough are placed in the same cluster. Any border point close enough to a core point is put in the same cluster as the core point. Noise points are discarded because they fail the defined user conditions.

Algorithm 2.3 Simplified DBSCAN algorithm

1: Classify the points as core, border or noise points
2: Remove noise points
3: Connect all core points that are within Eps
4: Create a separate cluster for each group of connected core points
5: Assign each border points to one of the clusters

Time and Space Complexity

The time complexity of the DBSCAN is O(number of points * time to find the point in the Eps-neighborhood). In the worst scenario, the time complexity can be $O(m^2)$. Nevertheless, in low-dimensional spaces, their data structures, that allow effective retrieval of all points within a predefined distance, and the time complexity can be $O(m \log m)$.

The space complexity is $O(m)$ because it is only required to keep a small quantity of data for each point. This data is the cluster label and the classification of each point as core, border or noise (Salkind 2013).

Strengths and Weaknesses

The biggest strength of density-based clustering can handle arbitrary shapes, sizes and noise points efficiently. One of the weaknesses of the DBSCAN algorithm is the difficulty of handling clusters with varying densities. Analysing high-dimensional data is another weakness because it is difficult to define such data using density approaches (Salkind 2013).

OPTICS Clustering

OPTICS (Ordering Points to Identify Cluster Structure) is a density-based algorithm (demonstrated by Ankerst et al. 1999), and the basics are similar to the DBSCAN algorithm. It adds two more concepts: (1) core distance - the minimum radius to classify a data object as core point, (2) Reachability Distance - is the smallest distance to the point that is density-reachable from a core point (Ankerst et al. 1999).

This algorithm was developed due to one of DBSCAN's biggest constraint: the problem of detecting meaningful clusters in data with varying and nested density. The OPTICS algorithm resolves this issue by ordering the points in the database such that spatially closest points become neighbours in the ordering. Each point stored in the database has the core-distance and a reachability-distance and these attributes are enough to extract all density-based clustering (Ankerst et al. 1999).

Strengths and Weaknesses

The biggest strength of density-based clustering can handle arbitrary shapes, sizes and noise points efficiently. The main weakness is the necessity of large computational requirements. Comparing to DBSCAN, this algorithm needs more memory as it maintains a priority queue to determine the next data point which is closest to the point currently being processed and more computational power due to complicated nearest neighbour queries (Gupta 2020).

Clustering algorithms comparison

The clustering algorithms are used to automate the process of analysing and understanding spatial data (Halkidi, Batistakis, and Vazirgiannis 2001). In this project, it is used to identify patterns that may exist in large spatial databases. To analyse spatial data in this context of facility location problem, it is essential to understand:

- The wide range of shapes of the produced clusters;
- The importance of identify outliers and noise;
- The definition of the coverage distance;
- The number of clusters, used as potential facility locations;
- The consumed time and space complexity (c.f. table 2.4).

The table 2.4 details the time and space complexity of the most relevant clustering techniques for this problem. The DBSCAN and the OPTICS have better time and space complexities. The worst technique, according to time and space complexity, is the K-means algorithm.

Table 2.4: Comparison between clustering algorithms

Algorithm	**Technique**	**Time Complexity**	**Space Complexity**
K-means	Centroid-based	$O(I * K * m * n)$	$O((m + K)n)$
DBSCAN	Density-based	$O(m \log m)$	$O(m)$
OPTICS	Density-based	$O(m \log m)$	$O(m)$

The table 2.5 details the main characteristics of the K-means, DBSCAN and OPTICS clustering algorithms. In order to handle geospatial data, the algorithm should be capable to analyse arbitrary shapes, outliers and noise, produce by itself the number of clusters according to a radius.

The algorithm should have the capability to search for high-density areas of points that are not necessarily globular, which is the case of the majority of the data that are analysed to identify potential facility locations.

The application to large spatial databases rises the following requirements for clustering algorithms: minimal requirements of domain knowledge to determine the input parameters, discovery of clusters with arbitrary shape and good efficiency on large databases (Ester et al. 1996).

Thereby, the K-means does not fit the solution, and the adequate techniques can be the DBSCAN and OPTICS. Thus, comparing the DBSCAN and OPTICS algorithms, the OPTICS requires more memory as it maintains a priority queue to determine the next data point which is closest to the point currently being processed and more computational power due to complex nearest neighbour queries. Another limitation of the OPTICS is that the technique does not segregate the given data into clusters (Gupta 2020).

Table 2.5: Comparison between main characteristics of clustering algorithms by Halkidi, Batistakis, and Vazirgiannis 2001

Algorithm	**Geometry**	**Outliers, Noise**	**Input Params**	**Results**
K-means	Non-convex shapes	No	Number of clusters	Center of clusters
DBSCAN	Arbitrary shapes	Yes	Cluster radius (*Eps*), minimum number of objects (*MinPts*)	Assignment of data values to clusters
OPTICS	Arbitrary shapes	Yes	Maximum Epsilon	Augmented ordering of the database representing its density-based clustering structure

Thus, the chosen algorithm is the DBSCAN due to the suitability to this problem, the maturity, the broad community support and documentation. The density based technique has been used by various researchers for, determining locations in urban areas for multiple facilities (Kant et al. 2018).

2.4 Decision Support System

A Decision Support System (DSS) is a computer-based system designed to help and support managers perform better business decisions. DSS requires structured data to be transformed into knowledge capable of helping the decision-makers. Before data become a part of a DSS it must be cleaned up by removing inconsistent and erroneous data, integrated by combining different data sources, selected by choosing the most relevant data, transformed, processed and evaluated to set up a well-designed user interface (Felsberger, Oberegger, and Reiner 2017). There are three different types of decisions: (1) structured, (2) semi-structured and (3) unstructured decision tasks.

Using structured decision tasks, the user is capable of understanding the result of the analysis of a large amount of data and relationships that are stable, however, limits the user's ability to process the aspects of the decision (Hosack et al. 2012). This kind of decision has a clear objective and well-defined input and output.

In the semi-structured and unstructured decision tasks, systems analyse a large amount of data and relationships but also attempt to alleviate the effect of some unknown parameters and relationships on the decision. (Hosack et al. 2012). In these scenarios, the decision-maker plays an important role to analyse and make a decision based on the provided information.

2.4.1 Types of Decision Support Systems

There are six main approaches of DSS: model-driven, data-driven, communication-driven, documentation-driven and, knowledge-driven. In this dissertation are introduced two types of DSS, according to the relevance to the problem's context.

Model-driven DSS

A model-driven DSS is used to support decision-makers in analysing a situation. In this type of DSS, models such as AHP, decision tree analysis or multi-criteria decisions, represent a simplified reality of a given problem. The goals are to find the most beneficial alternative in a given context (Felsberger, Oberegger, and Reiner 2017).

Data-driven DSS

A data-driven DSS is typically separated by feature and provides access to and use of structured data and can maintain time-series data and real-time data. This type of DSS necessitates the availability of a large set of structured data (Felsberger, Oberegger, and Reiner 2017).

2.4.2 Decision Making Process

The first goal of the decision making process is to identify the goals, objectives and values that fit in the problem. This process has multiple phases. The first step before initializing a decision-making process is to identify the problem and the decision-makers associated with this process. The figure 2.5 shows the decision-making process designed and improved by Felsberger, Oberegger, and Reiner 2017.

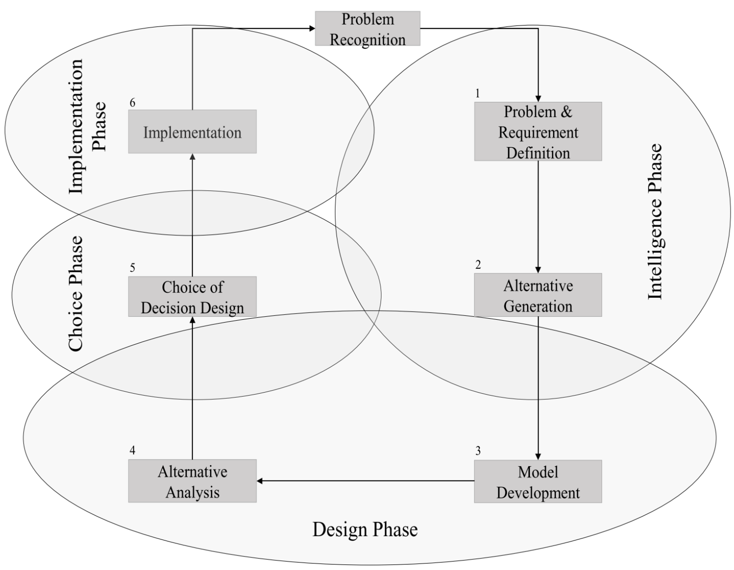


Figure 2.5: The decision-making process by Felsberger, Oberegger, and Reiner 2017

The Problem Requirement Definition and Alternative Generation are related to the Intelligence phase consisting in identifying and formulating the problem, and results in a decision statement. The problem requirement definition describes the initial problem statement and desired conditions, define the requirements of the problem to build a list of objectives. The alternative generation requires the creation of alternative solutions (Felsberger, Oberegger, and Reiner 2017).

The Design phase consists in the definition of the Model Development and Alternative Analysis - consisting in the development of alternatives and connect the decision-model with the objectives. The model development is used to analyse predefined alternatives measuring the effectiveness of the alternative. The developed model to compare alternatives should achieve the goals more efficiently. The next step is to evaluate the purposed alternatives against the requirements. To do this evaluation it is possible to use multiple approaches: Pros and Cons Analysis, AHP or TOPSIS. Finally, there is also the validation of solutions against the problem statement to ensure the resolution of the identified problem (Felsberger, Oberegger, and Reiner 2017).

Finally, the Choice of Decision Design and Implementation consist in the selection of the developed alternatives in the Design Phase. The result of this choice is a implementable decision or model in the Implementation Phase. The solution should fulfil the predefined needs in the problem recognition phase and achieve the established goals (Felsberger, Oberegger, and Reiner 2017).

Multi Criteria Decision Analysis

There is a multi criteria decision problem, when the decision maker has to make a decision between at least two possibilities of action with conflicting objectives (Frazão et al. 2018). MCDA is applied to support multiple complex decisions when the decision maker has various alternatives and have to chose one among them. A common decision making process follows the following steps (*MCDA* 2011):

1. Definition of the goals, interests and alternatives;
2. Identification of the decision makers;
3. Design and build a decision framework;
4. Rate alternatives based on a relative scale or ordinal scale. In the relative scale, the decision maker rates each alternative regarding to others in satisfying a particular interest. In the ordinal scale, the decision maker assigns to each alternative a rating for how well it satisfies a particular interest.
5. Weight stakeholder interests;
6. Score the alternatives;
7. Decision among the alternatives and produced scores.

The DSS helps the decision maker focus in what is important in a consistent and logical way. The MCDA is used for (1) dividing the decision into understandable components, (2) analysing each component, and (3) integrating the components to produce outcomes to be analysed by the decision makers.

Technique for Order of Preference by Similarity to Ideal Solution

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) (Hwang and Yoon 1981) is a method of the multi criteria decision making. The principle used by the method is to chose the alternative with the shortest distance from the positive ideal solution from a geometrical point of view using the Euclidean distance to determine the relative proximity of an alternative with the optimal solution and the farthest from the negative-ideal solution (Widianta et al. 2018).

In summary, the method is based on the concept of measuring distance from two hypothetical solutions: negative-ideal solution (NIS) and positive ideal solution (PIS) (Panda and Jagadev 2018). The PIS is the one that maximizes the benefit criteria and minimizes the cost criteria, while the NIS is the opposite (Aliyeva 2018). The steps of TOPSIS are represented in the Figure 2.6.

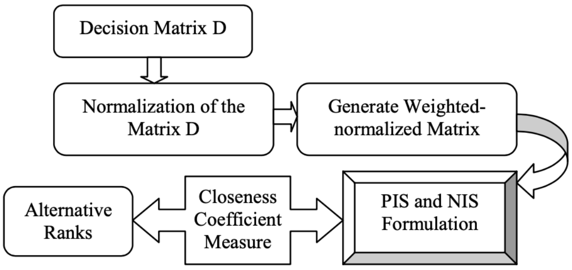


Figure 2.6: TOPSIS procedure (Panda and Jagadev 2018)

The first step is the determination of the weight of criteria and construction of the decision matrix. The second step is the calculation of the normalized decision matrix (Jozaghi et al. 2018).

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m}(x_{ij})^2}}$$

The third step is the calculation of the weighted normalized matrix by multiplying each element by the weight. Next, it is necessary to identity the ideal best value and the ideal worst value. The calculation of the separation of each alternative from the positive ideal solution and the negative ideal solution can be done by using the next equations (Jozaghi et al. 2018).

$$P_{i+} = (\sum_{j=1}^{n}(v_{ij} - v_{j+}^2))^{1/2}$$

$$P_{i-} = (\sum_{j=1}^{n}(v_{ij} - v_{j-}^2))^{1/2}$$

Next step is the calculation of the relative closeness to the positive ideal solution (Jozaghi et al. 2018):

$$C_i = \frac{S^-}{S^+ + S^-}$$

Finally, the definition of the rank of the alternatives according to the previous closeness results (Jozaghi et al. 2018).

Analytical Hierarchy Process

The Analytical Hierarchy Process (AHP) was introduced by Thomas Saaty (Saaty 1987) and this model is an effective tool to deal with complex decision making process, and may aid the decision maker to set priorities and guide it to the best decision (Mocenni 2011). Using a series of pairwise comparisons, and then synthesizing the results, the AHP helps to capture both subjective and objective aspects of a decision. In addition, the AHP includes a technique for checking the consistency of the decision maker's evaluations, thus reducing the uncertainty in the decision making process (Mocenni 2011). The main steps in problem solving using the AHP are the: decomposition, comparative judgements and synthesis of priorities (Saaty 1987).

The decomposition step (c.f. figure 2.7) is related to construction of a simple problem with elements in a level being independent from those in succeeding levels, so the objective is to decompose the problem from the more general to the more specific and concrete. The objective is to define the goal of the problem, the criteria in the second level and the alternatives for each criteria (Saaty 1987).

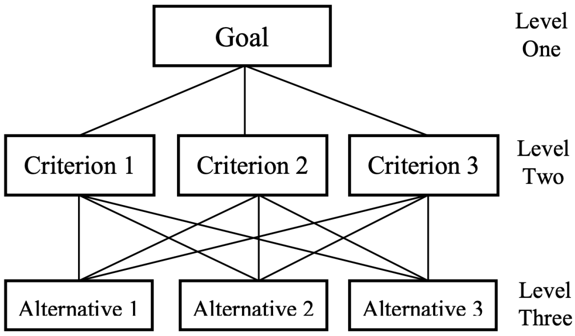


Figure 2.7: Hierarchical AHP Tree (Taherdoost 2018)

The comparative judgements is used to establish priorities among the parameters of the hierarchy based on a series of judgements using pair wise comparisons resulting in a pair wise comparison matrix. The parameters are rated by a preference level using a scale from 1 to 9 as shown on table 2.6 (Shukla, Upadhayay, and Dhamija 2019).

Table 2.6: Score of importance of variable (Saaty 1987)

Importance Scale	Definition of Importance Scale
1	Equally Important Preferred
2	Equally to Moderately Important Preferred
3	Moderately Important Preferred
4	Moderately to Strongly Important Preferred
5	Strongly Important Preferred
6	Strongly to Very Strongly Important Preferred
7	Very Strongly Important Preferred
8	Very Strongly to Extremely Important Preferred
9	Extremely Important Preferred

After assign a value to each pair wise comparison it is necessary to normalise the pair-wise comparison matrix by dividing each element by the sum of the column. Next, the criteria weights are calculated by determining the average value of each row. The consistency is given by the non-normalised value multiplied by the correspondent criteria weight. Next, the weighted sum value is given by the sum of each row and the ratio weights is given by the division of the weighted sum value by the criteria weight. To calculate the consistency index it is necessary to calculate the $\lambda\,max$ value by calculating the average ratio weight. The consistency index (C.I.) formula where n is the number of compared alternatives:

$$CI = \frac{\lambda max - n}{n - 1}$$

Finally, the consistency ratio is calculated by the next formula. The random index is given by the value of the random consistency index presented in the table 2.7.

$$ConsistencyRatio = \frac{ConsistencyIndex}{RandomIndex}$$

Table 2.7: Random Consistency Index (Taherdoost 2018)

Dimension	**RI**	**Dimension**	**RI**
1	0	6	1.2358
2	0	7	1.3322
3	0.5799	8	1.3952
4	0.8921	9	1.4537
5	1.1159	10	1.4882

The synthesis of priorities is used to define a set of overall priorities for the hierarchy. The decision maker can compare the qualitative and quantitative information using the informed judgments to derive weights and priorities for checking the consistency of the judgments. The last step is regarding the selection of the best alternative (Shukla, Upadhayay, and Dhamija 2019).

Multi Criteria Decision Making techniques comparison

The application of multiple criteria decision making (MCDM) techniques has increased in the last years. The need to build advanced decision models that can support decision-makers promotes the integration of MCDM techniques with systems. AHP and TOPSIS are the most used MCDM techniques to resolve facility location problems (Jozaghi et al. 2018). The table 2.8 shows the advantages and disadvantages of each technique and both methods are applicable to this dissertation.

To identify the most suitable method to this dissertation, it is considered the study developed by Jozaghi et al. 2018 which focus on a comparative analysis of TOPSIS and AHP in the context of dam site selection with geospatial and external factors. The results of the study show that TOPSIS is a more appropriate technique to facility location problems than AHP method. Other comparative study of Multi-Criteria Decision Support Methods (Widianta et al. 2018), collects the option of researchers between four methods (AHP, TOPSIS, SAW (Kaliszewski and Podkopaev 2016) and PROMENTHEE (Mareschal et al. 2005)) and the results shows that the TOPSIS method is the preferred one.

Table 2.8: Comparison between AHP (Source: Aruldoss, Lakshmi, and Prasanna Venkatesan 2013) and TOPSIS (Source: Panda and Jagadev 2018)

MCDM Method	Advantages	Disadvantages
AHP	1. Flexible, intuitive and checks inconsistencies; 2. Consistent decision making.	1. Irregularities in ranking; 2. A high number of criterion means more pair wise comparisons.
TOPSIS	1. Method focused to decision maker preference and choice; 2. Clear differentiation of all alternatives; 3. Easy decision making; 4. Applicable to qualitative and quantitative data.	1. The preliminary issue of the method is with the normalized decision matrix operation, where the decision maker assign a random value for the criteria and alternatives.

2.5 Technology

This section only focuses on the essential machine learning libraries for working with geospatial data. Their purpose is to facilitate the complicated data analysis process. The premise of this study is that the base technology (e.g. Python or R) need to be used to develop a REST API and be capable of applying clustering techniques (DBSCAN). For that reason and due to the stability and size of the community of the technology, this work uses Python. Python is used to develop the cluster analysis subsystem, and it is responsible for analysing the telemetry data to identify the most relevant clusters. The next subsections detail the most relevant machine learning frameworks.

2.5.1 Scikit-learn

Scikit-learn is a library that features a host of the classical machine learning algorithms. It includes options for both supervised and unsupervised learning. The dissertation focuses on unsupervised learning and considering this the library has a module (*sklearn.cluster*) dedicated to clustering and supports a variety of clustering algorithms such as K-Means, DBSCAN and OPTICS detailed in this work (*scikit-learn* 2020).

Some of the strengths are the fast learning curve, the native support of the most relevant machine learning algorithms and the stability and its support community. Regarding the weaknesses, does not support artificial neural network and GPU computing.

2.5.2 Shogun

Shogun is an open-source general-purpose ML library that offers a wide range of efficient and unified ML methods (*Shogun Machine Learning* 2020). The idea behind this framework is that the underlying algorithms are transparent, accessible and free to use (Nguyen et al.

2019). This framework has native support to clustering techniques and some implementations ready to use.

Some of the strengths are the support for many languages such as Python, R, Java/Scala, C# and Ruby. The high number of ML algorithms. Regarding the weaknesses, there is a lack of documentation.

2.5.3 TensorFlow

TensorFlow is an open-source machine learning project with support to all platforms, developed and maintained by Google, that uses a single data flow graph to achieve excellent performance (Shatnawi et al. 2018). According to the official website, it has a comprehensive, flexible ecosystem of tools, libraries and community resources that allows developers quickly build and deploy ML-powered applications (*TensorFlow* 2020).

Some of the strengths are the flexibility, ready to use ML models, scalability and large community. Regarding the weaknesses, the learning curve is slow.

2.5.4 PyTorch

PyTorch is an open-source machine learning framework that accelerates the path from research prototyping to production deployment. This technology was created by Facebook Artificial Intelligence Research team and developed using Python, C++ and CUDA. This technology is flexible and provides high-level efficiencies and speed (*PyTorch* 2020).

Some of the strengths are the learning curve, the support to dynamic graphs and the support of GPU acceleration. Regarding the weaknesses, this technology is growing, but its community is smaller than other frameworks, and this leads to fewer resources available online.

2.5.5 Technologies comparison

The technologies to develop machine learning techniques and to apply algorithms is very used nowadays due to the increasing complexity of the problems. There is a considerable amount of existing technologies suitable for ML and, this project focuses on four tools already detailed on the section 2.5: Scikit-learn; TensorFlow; PyTorch; Shogun.

To develop this project, the framework needs to have build-in solutions for unsupervised learning - clustering techniques, more precisely to have support to density techniques. The table 2.9 shows that the technologies suitable for clustering for this project are the Scikit-learn and Shogun.

Table 2.9: Comparison between technologies for Clustering Analysis System

Framework	Version	Learning Curve	Suitable for Clustering
Scikit-learn	0.23.2	LOW	YES
TensorFlow	2.1	HIGH	NO
PyTorch	1.4	LOW	NO
Shogun	6.1.4	MEDIUM	YES

Analysing the implemented clustering algorithms in each framework, the Shogun does not have support for density-based clustering techniques. Meaning the only framework with built-in solutions for clustering with DBSCAN is the Scikit-learn. Furthermore, the learning curve is low, the GitHub community is large (used by 142.345 people - mid 2020) and the available resources and support are extensive. One of the drawbacks of the Scikit-learn is the immaturity to be used in a production environment.

Table 2.10: Snapshot of GitHub community and analysis from Codacy (Nguyen et al. 2019)

Framework	**Contributors**	**Commits**	**Code Quality [A-F]**
Shogun	174	17569	A
Scikit-learn	1873	26124	B

The table 2.10 details the GitHub community and the code quality using Codacy in 2019, reveling a high number of total commits and the code quality is classified in the second grade of six.

2.6 Summary

In summary, this chapter concerned the function of describing the highest level of development or scientific field at a particular time. Also has comparative studies for the most relevant subjects of this project.

First, the problem and its context were detailed. The expected telemetry data structure was set (c.f. section 2.1.1) and the most relevant factors were described (c.f. section 2.1.2).

Second, state of the art about the facility location problems was described to contextualise the scope of the problem (c.f. section 2.2): (1) Single Facility Location Problem; (2) Covering Problem; (3) Center Problem; (4) Location Allocation Problem.

Third, Machine Learning was described, and its learning methods were studied: supervised learning, reinforcement learning and unsupervised learning. This project focuses on unsupervised learning.

In section 2.3.4, were described as more in-depth the main concepts of clustering, various approaches to divide data into clusters, different types of clustering and clusters, and the essential clustering techniques. The clustering algorithms (K-Means, DBSCAN and OPTICS) were compared, and the DBSCAN algorithm was chosen for this project. The clustering algorithm selection is correlated with the selection of the technology to perform the clustering analysis (c.f. section 2.5).

Finally, the decision making process (c.f. section 2.4.2) details the different techniques that can be used to produce facility location suggestions based on alternatives and criterion. After analysing the comparison of the MCDA techniques, it was chosen the TOPSIS algorithm.

Chapter 3

Value Analysis

The Value Analysis concerns the function of a product to meet the demands or application need by a customer.

The value was analysed based on the: Innovation Process using the New concept development (NCD) model by P. Koen et al. 2001; calculation of the TAM, SAM and SOM; value of the solution for the customer and the perceived value; definition of the value proposition; Business Canvas Model; Quality function deployment (QFD). This section explores those components in detail.

3.1 Innovation Process

The technological evolution brings rapidly new products to the market, as well increases the need and demand for products with quality to fulfil the market needs and desires. The product quality is fundamental for the prospect of the solution and to the business growth. So to guarantee the development of a valuable product, it was defined a method with multiple steps characterizing the innovation process, demonstrated in Figure 3.1.

The process of innovation is divided into three main steps: Fuzzy Front End, New Product Development and commercialization.

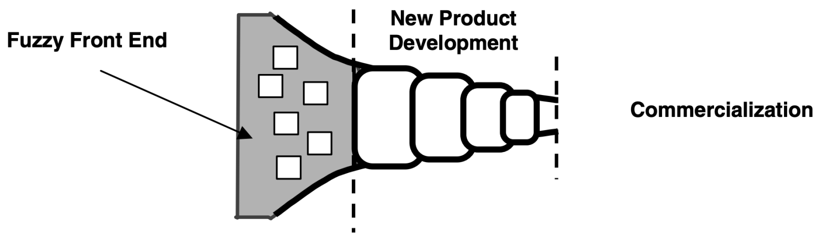


Figure 3.1: Innovation Process by P. A. Koen, Ajamian, et al. 2001

The Fuzzy Front End is the process of exploring new opportunities and usually is used the New Concept Development model to describe these steps of the Fuzzy Front End (P. Koen et al. 2001). The NCD model provides a common language and definition of the key components of the Front End of Innovation (P. Koen et al. 2001).

The NCD model has three main components: the engine, the Fuzzy Front End elements and the influencing factors (c.f. figure 3.2). The engine is regarding the main aspects of the business (leadership, business culture or strategy) that drives the company's strategy. The five Fuzzy Front End elements are regarding the opportunity identification, the opportunity analysis, the idea genesis, the idea selection, the concept and the technology development. Finally, the influencing factors are unpredictable and regarding events occurring outside the business scope (P. A. Koen, Ajamian, et al. 2001).

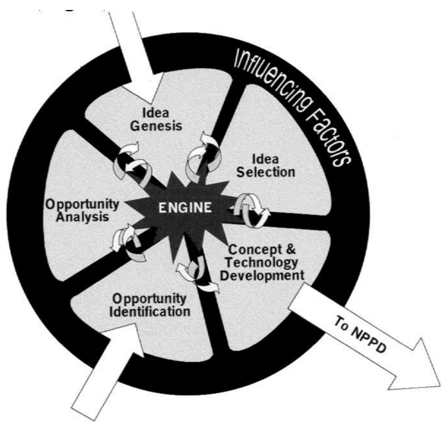


Figure 3.2: The New Concept Development Model by P. Koen et al. 2001

The NCD model consists of five front end elements: opportunity identification, opportunity analysis, idea genesis, idea selection and concept and technology development. These five concepts are described below to define the NCD model according to this project.

3.1.1 Opportunity Identification

Facility location problems are complex and decisive strategic challenges. Siting facilities are based and long-lasting, impacting numerous operational and logistical decisions. Furthermore, property acquisition and facility construction make facility location or relocation projects long-term investments (Owen and Daskin 1998). Facility location problems locate a set of facilities to minimise the cost of a set of demands taking consideration of some constraints (Farahani and Hekmatfar 2009). The decision-makers must select sites that will perform well for the facility's lifetime. It is imperative to the decision-makers to make better strategical decisions.

The companies using fleet management systems to track and analyse their fleet, they have persisted data from their assets since they subscribed to the service. This data can be analysed to produce sustained suggestions of geospatial locations for a facility location and help them to make better decisions in a location-allocation problem.

The development of a decision support system can help the business owner to make better decisions. The identification of clusters can accomplish this through the analysis of the data consumed by the fleet management system and by the selection of the most relevant factors that can affect their business. The produced suggestions can support the company in the process of this critical decision by analysing a complex problem with numerous factors in seconds, simplifying the whole process by the evaluation of existing data.

In fleet management market, the automotive sector has the most significant share and according to UN Comtrade[1], around 218 million light and heavy commercial vehicles were traded back in 2014, and this number is increasing every year. These assets are associated to companies with potential interest to have a fleet management solution to track their assets.

[1] https://www.alliedmarketresearch.com/fleet-management-market

It is not guaranteed to have the will to subscribe to this DSS but can be an advantage comparing to other solutions and a key player in the process of negotiation.

TAM, SAM & SOM

The Total Available Market (TAM) is the term used to reference the revenue opportunity available for a product or service. The TAM is the amount of annual revenue, expressed in euros per year, that the business would earn if achieved 100% market share (MIT 2012). It is essential to develop a conservative, defensible number to define the TAM. For this project, one approach is to define the TAM as the worldwide fleet management market. However, it is an unreal number due to insufficient resources and impossibility to satisfy such a big market (MIT 2012).

For this project, the TAM is focused on the United Kingdom market share according to the current company position on the market. In 2019, there were 5,869,000 businesses in the UK (c.f. table 3.1), this value is increasing year by year. Compared to 2018, were created more than 200,000 new companies (W., Davies, and Yoder 2019).

Table 3.1: Businesses by industry in the UK, 2019 (W., Davies, and Yoder 2019)

Industry	**Number of Businesses**
Agriculture, mining and utilities	194,000
Manufacturing	276,000
Construction	1,037,000
Retail	547,000
Transportation	360,000
Accommodation & food	202,000
Information and Communications Technology	370,000
Financial and Insurance	91,000
Real state activities	113,000
Professional and scientific	868,000
Administrative & support service	512,000
Education	307,000
Health and social work	361,000
Arts and recreation	290,000
Other services activities	340,000
Total TAM	5,869,000

This service is an extension of the current Fleet Management Solution, and so the price is fixed according to the size of the company (c.f. table 3.2) and paid per use. To calculate the annual revenue of the TAM, it is necessary to define the service's price. The value is calculated based on:

$$TotalAnnualRevenue = \sum_{j=1}^{n}(NumberofBusinesses * Usage * Price)$$

According to the calculations on the table 3.2, the TAM focus on 5,869,000 businesses and the total annual revenue is €30,127,500.

Table 3.2: TAM - Annual revenue in the UK by number of employees (W., Davies, and Yoder 2019)

Employees	**Businesses**	**Price (€)**	**Total TAM (€/year)**
No employees	4,458,000		
0-9	5,613,000	5	28,065,000
10-49	211,000	7.5	1,582,500
50-249	36,000	10	360,000
250+ employees	8,000	15	120,000
Total Annual Revenue	5,869,000		30,127,500

The Serviceable Available Market (SAM) is the segment of the TAM targeted by the service within the geographical reach (MIT 2012). The service is focused on companies that have the necessity and capability of building or buying a new facility location, and so the SAM is focused on companies with more than ten employees (c.f. table 3.3).

Table 3.3: SAM - Annual revenue in the UK by number of employees (W., Davies, and Yoder 2019)

Employees	**Businesses**	**Price (€)**	**Total SAM (€/year)**
10-49	211,000	7.5	1,582,500
50-249	36,000	10	360,000
250+ employees	8,000	15	120,000
Total Annual Revenue	255,000		2,062,500

According to the calculations on the table 3.3, the SAM focus on 255,000 businesses and the total annual revenue is €2,062,500.

The Serviceable Obtainable Market (SOM) is the portion of the SAM that realistically get to use the service (MIT 2012). In the next three years, the Fonix Telematics is focused on selling the Fleet Management Solution to companies with more than 250 employees. This extension adds value to the current solution and can help the companies to grow using their existing telemetry data. So, the SOM focus on 8,000 businesses and the total annual revenue is €120,000.

It is important to refer that due to lack of available information and data, it was not possible to make a correlation between the number of employees, the business industries and the number of assets per business.

3.1.2 Opportunity Analysis

According to Mordor Intelligence, the fleet management solutions market was valued at EUR 5.5 billion in 2019 and is expected to reach a value of EUR 16.6 billion by 2025, at a Compound annual growth rate (CAGR) of 20.07% over 2020-2025 (*Fleet Management Solutions Market | Growth, Trends, and Forecast (2020-2025)* 2019).

The fleet management market is competitive (c.f. figure 3.3) and requires innovation to increase the number of customers that subscribed to the service and to increase the market value of the solution.

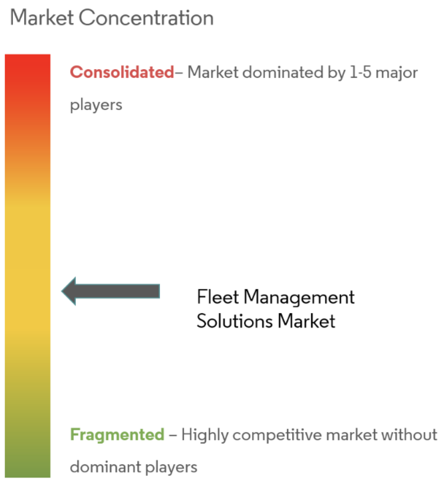


Figure 3.3: Market Concentration by *Fleet Management Solutions Market | Growth, Trends, and Forecast (2020-2025)* 2019

According to Mordor Intelligence study, currently, the largest market is in Europe and the inclusion of the location-allocation feature is considered a distinctive factor, capable of being incorporated in the value of the offered solution.

3.1.3 Idea Genesis

An idea is the first state of a new service (P. A. Koen, Bertels, and Kleinschmidt 2014), this idea is based in need of a customer to expand their geospatial coverage and the request of an analysis of their assets journeys to understand the most valuable locations for them. To understand and discuss the idea, it was necessary to define a series of questions as follows:

- How to make use of the existing telemetry data and produce value for the client?
- How to simplify the decision-maker decision in facility location problems?
- How to increase the value of the fleet management base system?

3.1.4 Idea Selection

In most businesses, there are so many product ideas that the critical activity is to choose which ideas to pursue to achieve the most business value (P. Koen et al. 2001).

Many ideas must be allowed to grow and advance with less certainty (P. Koen et al. 2001), but to select the idea, it is important to answer the questions in section 3.1.3. Based on this statement, the potential of the idea was analysed, and the conclusion was that this could be a product that can increase customer satisfaction and the company's rate of innovation to capture new clients.

How to make use of the existing telemetry data and produce value for the client? The client assets are transmitting data to the system and have been persisted since then. It is possible to develop several ideas to make use of this data, and the main idea is to help the decision-maker to make batter decision.

How to simplify the decision-maker decision in facility location problems? Analyse the existing telemetry data and build suggestions to be used in the multi-criteria decision problem. Analyse the most relevant factors to the business and produce results based on the decision-maker preferences. The system simplifies the process by taking care of all constraints and alternatives.

How to increase the value of the fleet management base system? Develop innovative features to add value to the base fleet management solution.

3.1.5 Concept and Technology Development

Develop a Decision Support System (DSS) based on two main aspects: Analysis of telemetry data to identify geospatial clusters for a particular context; Analysis and identification of the most important factors that could affect a location-allocation problem.

Based on its business, the manager can choose the factors that are important to the growth of the company. The system produces a set of facility location suggestions for each identified cluster with a calculated rate based on the chosen factors (c.f. table 2.3).

3.2 Value

Value has in recent years, become a popular term in management theory and practice in general as well as in economic theory and architectural management. Value has different meanings depending on the context it is applied, such as economic, use or perceived value. (Jensen 2012). The value can be a combination of quality, service and price that reflects the perceived tangible and intangible benefits and costs (Kotler and Keller 2006).

The perceived value can be regarded as the relation between the perceived benefits and perceived sacrifices. Different customers segments can have different value perceptions about the same product or service (Ulaga and Eggert 2006). If the benefits have more weight than the sacrifices, the solution has value to the customer.

The value of this solution is the decision support system that helps the customer in the facility location decision, reducing costs and risks associated with this long term investment. To evaluate the value for the customer was created the table 3.4 with the benefits and sacrifices associated with this project.

The system is capable of providing suggestions and helping the customer through this complex problem of facility location-allocation problem. The customer doesn't need to analyse manually their client locations, their most common journeys or other factors that may affect the business strategy.

Table 3.4: Benefits and Sacrifices to the customer

Benefits		
Attributes	**Outcomes**	**Sacrifices**
Suggestions with the most adequate facility locations	Reduce costs related to fuel and assets	System's cost
Simplify complex strategic decisions by the use of existing data	Reduce risks	Adaptation to the system
	Reduce mileage	
	Increase operations coverage	
	Potential expansion of the number of customers	

The customer takes advantage of a system capable of analysing their whole journeys and build suggestions based on data analysis. This process automates and simplifies the customer decision process.

The most significant benefit of the solution is the ability to produce suggestions to business managers make better long-term strategic decisions based on the company needs, and the customer demands. Other benefits are the functional benefits, logistical benefits and the long-term results for the customer, whereas the sacrifices are mainly related to the cost of the service usage.

3.3 Value Proposition

A value proposition refers to the value a company promises to deliver to customers should they choose to use their services or products (Twin 2019).

This solution is for customers that want to relocate or create a facility for their operations. When a company is interested in creating or relocating a facility location, need to be aware of their business scope and all the related restrictions. It is imperative to analyse their client locations, journeys and be aware of the factors that may affect their geographical decision.

The decision support system is a service that produces facility location suggestions according to the business. This service can help the decision-makers to accomplish better decisions and potentially increase fleet performance, productivity and reduce operating expenses based on their fleet data.

A better decision help the company grow and focus on strategic business planning. This service takes advantage of the persisted data through the time and hides the complexity of the solution presenting a simplified version to the decision-maker.

3.4 Canvas Model

The solution helps the customer to make better long-term decisions for their business by reducing costs and risks associated with this complex decision of a facility location problem. The innovation and newness of the solution is another factor that may interest the customer.

The Canvas Model (c.f. figure 3.4) was developed to capture and describe the customer perspective and business perspective about the solution.

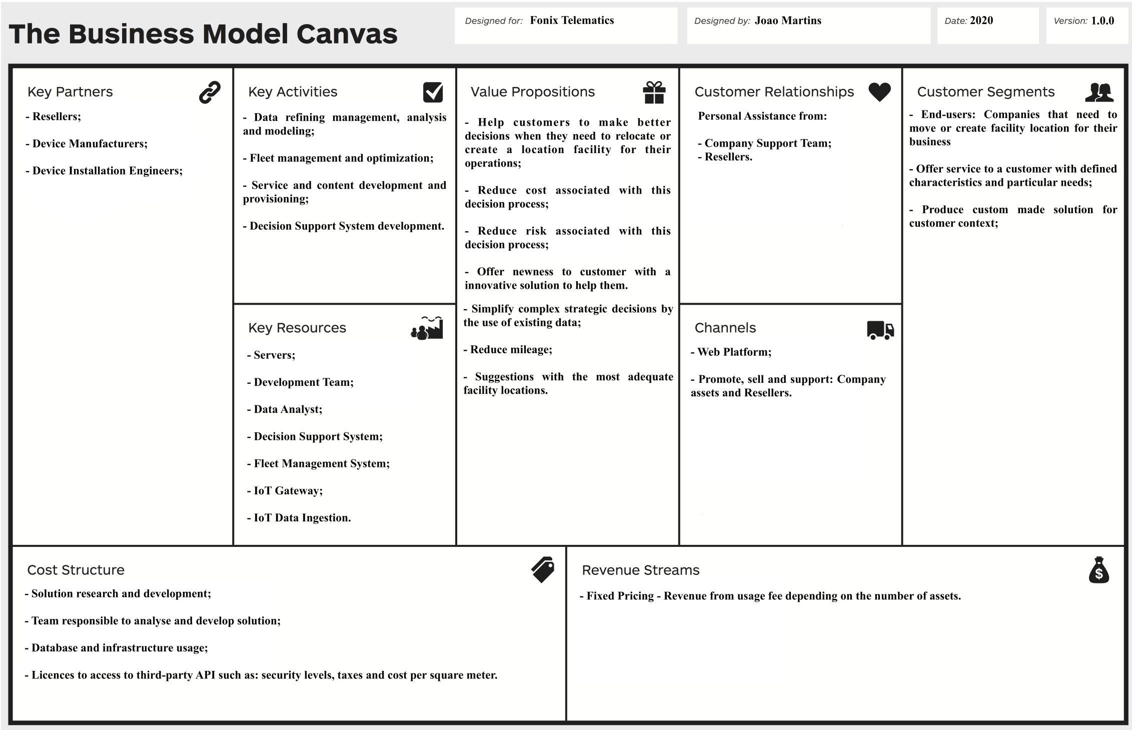


Figure 3.4: Business Model Canvas (*Business Model Canvas* 2020)

The front stage is regarding the customer perspective - Customer Relationships, Customer Segments, Channels and Revenue Streams. The customer segments are related to the companies that want to move or create an operations facility. In order to build a customer relationship, there is personal assistance from the resellers and the technical support team. The channels are the web platform and the resellers (companies that sell the fleet management solution). Finally, the revenue stream is a usage fee that varies according to the number of assets.

The Value Proposition belongs either to the customer and business perspective. This system wants to help customer to make better decisions, reduce costs and risks associated with the decision process, offer an innovative service that is capable of analysing their data and produce facility location suggestions.

The backstage is regarding the business perspective - Key Partners, Key Activities, Key Resources and Cost Structure. The key partners for this service are the resellers, the device manufacturers and the device installation engineers that ensure that the service is correctly delivered. The key activities are related to the correct data analysis, the continuous fleet management optimisation, the assistance and support of the service and the development of the decision support system. The key resources are the required resources to ensure successful service quality and delivery. Finally, the cost structure is related to the research, the development and analysis team, the infrastructure costs and third-party licenses.

3.5 Quality Function Deployment

Quality Function Deployment (QFD) is a system for designing a product or service based on customer demands as well the functionalities to identify the requirements (Warwick Manufacturing 2007). The purpose of QFD is divided into three (Warwick Manufacturing 2007): (1) Launch products or services to market faster and at a lower cost; (2) Achieve customer-driven product design; (3) Provide a tracking system for future process improvements.

A House of Quality (*QFD Online* 2019) synthesizes the relationship between customer requirements and functional requirements (Warwick Manufacturing 2007).

In order to capture the customer satisfaction it is necessary to fulfil the customer requirements. This service focuses on the following requirements:

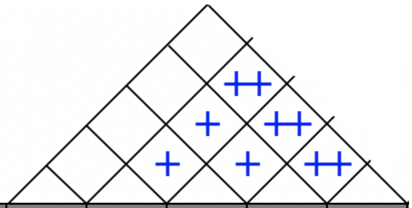
- User security (Weight: 15.0);
- Support facility location decision by providing suggestions (Weight: 35.0);
- Automate geospatial data analysis (Weight: 15.0);
- Reduce decision risks and costs (Weight: 25.0);
- Ease of use (Weight: 10.0).

To meet customer requirements, it is necessary to define a strategy and a set of requirements to fulfil their needs. So, according to the quality characteristics were defined by the following functional requirements:

- Authentication and Authorization service;
- Telemetry data analysis;
- Suggestions with a decision support system;
- Fetch factors data from reliable external resources;
- Multi-criteria decision analysis.

These requirements were ranked (0 = Easy, 10 = Extremely Difficult) and there was defined a target value, in weeks, that is relative to the projected time to complete a requirement.

The House of Quality is a QFD design tool that wants to synthesize the quality and the potential of the system to the customer. The figure 3.5 was developed according to the project scope.



				Column #	1	2	3	4	5
				Direction of Improvement: Minimize (▼), Maximize (▲), or Target (x)	X	X	X	X	X
Row #	**Max Relationship Value in Row**	**Relative Weight**	**Weight / Importance**	**Quality Characteristics** (a.k.a. "Functional Requirements" or "Hows") / **Demanded Quality** (a.k.a. "Customer Requirements" or "Whats")	Autentication and Authorization service	Telemetry data analysis	Suggestions with decision support system	Fetch factors data from government data sources	Muti criteria decision analysis
1	9	15,0	15,0	User Security	Θ				
2	9	35,0	35,0	Support facility location decision		Θ	Θ	Θ	Θ
3	9	15,0	15,0	Automate geospatial data analysis		Θ		O	
4	9	25,0	25,0	Reduce decision risks and costs		O	Θ	O	Θ
5	3	10,0	10,0	Ease of use			O		
6									
7									
8									
9									
10									
				Target or Limit Value	1 weeks	3 weeks	3 weeks	2 weeks	2 weeks
				Difficulty (0=Easy to Accomplish, 10=Extremely Difficult)	3	6	8	5	6
				Max Relationship Value in Column	9	9	9	9	9
				Weight / Importance	135,0	525,0	570,0	435,0	540,0
				Relative Weight	6,1	23,8	25,9	19,7	24,5

Figure 3.5: QFD - House of Quality

3.6 Summary

In summary, value analysis concerns the function of a product to meet the demands or application need by a customer.

The Innovation Process is composed by the Opportunity Identification with a detailed TAM, SAM and SOM analysis, the Opportunity Analysis, the Idea Genesis, the Idea Selection and the Concept and Technology Development.

The Value of this solution is the decision support system that helps the customer in the facility location decision, reducing costs and risks associated with this long term investment.

The Value Proposition was set as a service responsible for producing suggestions that help business managers to make better decisions in a location-allocation problem. A better decision help the company grow and focus on strategic business planning. This solution takes advantage of the persisted data through the time and hides the complexity of the solution presenting a simplified version to the decision-maker.

The Canvas Model details customer and business perspectives. Finally, the Quality Function Deployment was described with the use of the design tool named as House of Quality.

Chapter 4

Design

The Design represents an architectural abstraction of the system that describes the system structure, how its elements work together and provide guidelines to support the development process.

This chapter details the following items: requirement analysis divided into functional requirements and non-functional requirements; system architecture with the proposed components and deployment views of the system; architecture and technologies for each subsystem; main flow execution of the solution.

4.1 Requirements Analysis

This section focuses on the actors that compose the solution, the functional requirements and the non-functional requirements.

4.1.1 Actors

This subsection details the role and purpose of the actors for each system. Despite the only existing actor on this solution is the Decision-Maker that is responsible for triggering the whole mechanism, it is described, for each system the purpose they have.

Decision Support System

This system is responsible for building facility location suggestions and retrieving them to the decision-maker. The main actor in this system is:

- Decision-Maker: Responsible for initiating the DSS flow execution by defining the input parameters of the mechanism. These parameters are related to the selection of the factors that the decision-maker considers more important to the facility location decision. This actor is also responsible for choosing facility location and list facility location decisions.

Geospatial Analysis System

The Geospatial Analysis System is responsible for analysing the telemetry data and producing geospatial clusters holding the potential facility locations.

- System: Responsible for gathering telemetry data from the database and apply a clustering algorithm to produce geospatial clusters to be used as alternatives in the multi-criteria decision analysis.

Facility Location Factors System

This system is responsible for retrieving a list of supported factors and the specific factor's data used as a criterion in the MCDA.

- System: Responsible for gathering data from external API to process and retrieve indicators and metrics according to the decision-maker preferences.

4.1.2 Functional Requirements

This section details the functional requirements for each subsystem. The figure 4.1 details the identified behavioral requirements through a use case diagram.

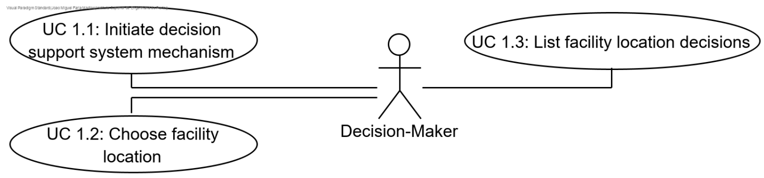


Figure 4.1: Use case diagram

Use Case 1.1: Initiate the decision support system mechanism

The authenticated and authorised decision-maker initiates the decision support system mechanism by defining the factors (c.f. table 4.1) that are more relevant to the facility location selection. The system receives the chosen factors and triggers the facility location suggestions process.

Table 4.1: Supported factors for facility location selection

Factor	Details/Metrics
Growth zones	Location evolution and investment
Construction Costs	Material prices, price per m^2 and wages
Indices of Deprivation	Employment Rank, Education and Skills, Health and Living environment
Prices per Square Foot	Prices per Square Foot
Council Tax Band	Value evaluation of the property
Safety levels	Crime rank per area

To produce ordered suggestions, it is necessary to make an MCDA composed of alternatives and criterion. The alternatives regard to identification of geospatial locations and the criterion regards to the data of the chosen factors for each identified location.

Use Case 1.2: Choose facility location

The authenticated and authorised decision-maker makes its facility location decision after getting and analysing the produced facility location suggestions.

Use Case 1.3: List facility location decisions

The authenticated and authorised decision-maker lists its previous facility location decisions.

4.1.3 Non-Functional Requirements

This section details the non-functional requirements for the system using FURPS+ composed by these quality factors: functionality, usability, reliability, performance, supportability and system constraints related to design, implementation, interface and physical constraints (Eeles 2004).

Functionality

The functionality focuses on describing the main system functionalities to be included in the solution (Eeles 2004). On the context of the security, the solution should have authentication and authorisation mechanism to ensure the user is authenticated and authorised to use the features provided by the system. The Fonix Telematics FMS already implements the system responsible for authenticating and verifying the authorisation. According to the localisation, the first version of the system only supports UK companies using the FMS.

Usability

The usability focuses on the user interface requirements. This quality factor is crucial to the solution due to the importance of the DSS. Considering this solution is an extension of the FMS solution, the user interface should be consistent through the company solutions. This factor needs to be considered efficient, understandable, operable, attractive, error preventive, learnable, accurate and effective (Shafinah et al. 2010). In summary, the system should be simple, straightforward, understandable and easy to use.

Reliability

The reliability focuses on availability, accuracy, and recoverability of the system (Eeles 2004). According to availability, the system must be available and functional even in failure scenarios. The system should be capable of creating new instances of the application in case of failure. In failure scenarios, the error should be contained in the application where the error occurred.

Performance

The performance focuses on the throughput of the system, system response time, recovery time, and start-up time (Eeles 2004). The mechanism of producing facility location suggestions is involved, and the response time should be acceptable, further investigation will detail this subject more precisely. In case of high throughput and load of the system, the infrastructure should be prepared to create new instances of the applications and to load balance the requests to reduce the instance load.

Supportability

The supportability focuses on the testability, extensibility, configurability, maintainability and adaptability (Eeles 2004). According to the testability, the solution should be tested by analysing the clustering metrics, the API metrics and the integration of the solution with different business scopes. The system should be extensible by allowing the addition of new factors and integration with other external factors providers. The distributed architecture contributes to the low coupling by segregating the responsibilities in a modular way, increasing the maintainability of the solution.

Design Constraints

The design constraints focus on the design limitations of the system (Eeles 2004). The design limitations are:

- NoSQL database responsible for persist telemetry data;
- The API uses REST architectural guidelines.

Implementation Constraints

The implementation constraints focus on resource limitations of coding or construction (Eeles 2004). The implementation constraints of the system were defined considering the main Fonix Telematics technologies and are the following:

- The DSS and Factors API should be implemented using .NET Core;
- The Geospatial Analysis System should be implemented using Python;
- The applications should be run in containers using Docker;
- The user interface of the DSS frontend should be implemented using Angular 8.

Interface Constraints

The interface constraints regard the interaction with external items (Eeles 2004). In this system, these external items are the external data providers to retrieve data for each factor. The first version of the system will use PropertyData API.

Physical Constraints

The physical constraints focus on the hardware used to hold the system (Eeles 2004). There are two essential constraints for this system because the infrastructure specification holding the API and the database are different. It is expected to the machine with the API have high CPU to handle the request throughput and the machine with the database instance must focus on I/O operations.

4.2 System Architecture

This section concerns on describing the system architecture regarding the logical and deployment views. The component view (c.f. section 4.2.1) has two alternatives that are compared, and the deployment view (c.f. section 4.2.2) has only one alternative that captures the overall of a simplified deployment structure.

4.2.1 Component View

This section details the component perspective and compares the different alternatives that can be applied to the solution. The main components regarding this solution are:

- Fonix Auth API: This API is responsible for authenticating and verifying the user authorisation;
- Frontend DSS: The user interface that is responsible for interact with the decision-maker to make its facility location decision;

- DSS API: This API is responsible for producing suggestions to expose to the decision-maker by the Frontend DSS. This system is also responsible for gathering data to build the MCDA to produce the ordered suggestions;
- Geospatial API: This API is responsible for analysing telemetry data and identify potential facility locations to be used by the DSS API as alternatives;
- Factors API: This API is responsible for maintaining a list of support factors and to communicating with services responsible for gathering factor's data. The way that the data is retrieved for each factor depends on the chosen architecture;
- External Factors: These API are consumed to retrieve factor's data, e.g., Data Government API, National Statistics API and Property Data API.

The alternative one was used to implement this project, more details about the decision can be found in the section 4.2.1.

Alternative one

The figure 4.2 details the first component view alternative for this solution. Regarding the advantages, this alternative stands for its architectural simplicity and can be used as an initial approach to developing a minimum viable product (MVP), satisfying early customers.

By contrary, the drawbacks are regarding the maintainability and deployment, as the solution growths the complexity of maintaining also increases, and the continuous deployment can be more complicated. As the Factors API growths due to increasing number of supported factors and integration with external services, it can often be challenging to maintain the application.

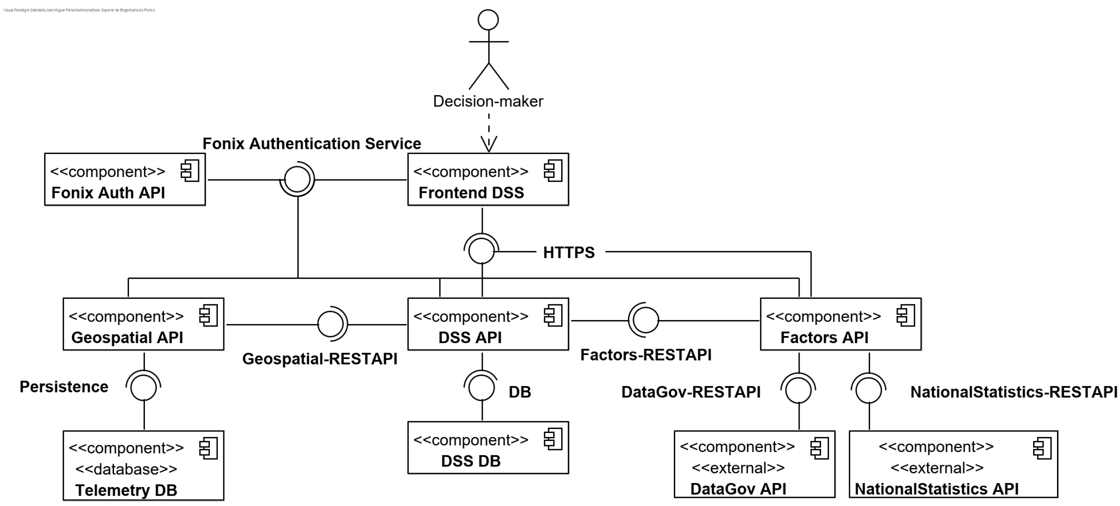


Figure 4.2: Component diagram - Alternative One

Nevertheless, this alternative was followed to develop the solution. Due to the context and uncertainty of the acceptance of the solution by the customers, this decision was made regarding the KISS principle. It is crucial to deliver an MVP to the customers to satisfy them in an early stage and collect relevant feedback to future changes.

Alternative two

The figure 4.3 details the second component view alternative for this solution. The main advantage of this alternative is the segregation of responsibilities of the Factors API. The Factors API continuous to exist but communicates with new API that is responsible for gathering data from external services. This division promotes API with specific purposes and responsibilities. By contrary, the drawbacks are regarding the architecture complexity and changes that span multiple services.

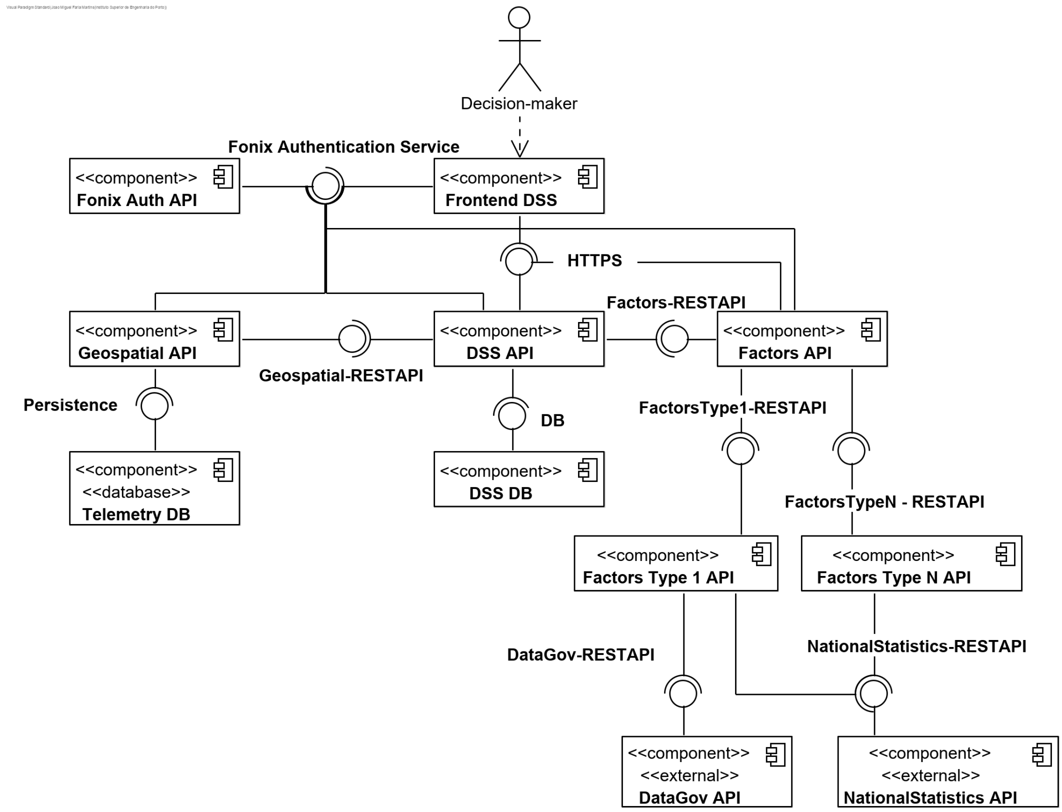


Figure 4.3: Component diagram - Alternative Two

4.2.2 Deployment View

The figure 4.4 details the deployment diagram of the solution, detailing the execution environment, the physical hardware and software of a system (*Deployment Diagram* 2020).

The components of the system are divided by the two different servers regarding different purposes and server configurations. Each server has a set of Docker containers running inside them. The MongoDB and PostgreSQL database is in the server named as Database Server, and they use TCP/IP protocol to communicate with other containers.

The DSS API, Factors API and Geospatial API (detailed in the sections 4.3, 4.5 and 4.4, respectively) are running in containers inside another server and they communicate through HTTPS. Finally, the DSS frontend is in the third server, also running inside a container. The DSS frontend is not a part of this project, and so, it is not implemented. Nevertheless, this component was included in the deployment diagram to support the future development of this component.

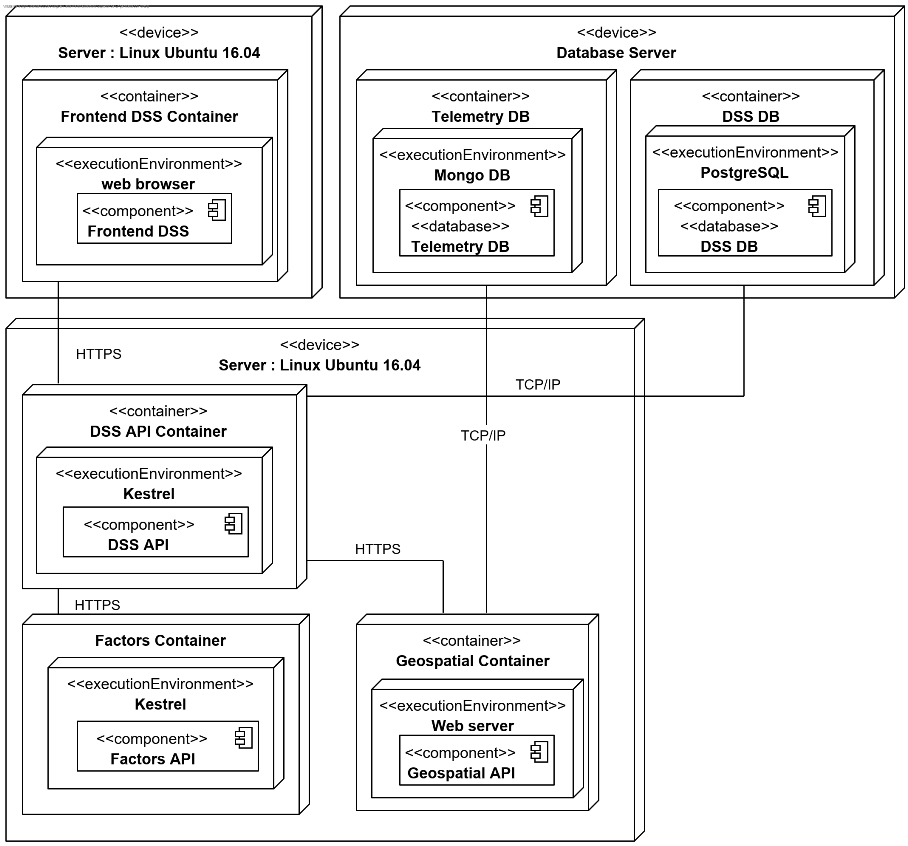


Figure 4.4: Deployment diagram

This deployment view is a proposal, and it is not guaranteed that this version would be the final version used by Fonix Telematics in a production environment.

4.3 Decision Support System

The DSS is responsible for building facility location suggestions and retrieving them to the decision-maker. This system has the following steps:

1. The decision-maker initiates the flow execution by defining the factors that are more relevant to the business;
2. The DSS invokes the Geospatial Analysis System (c.f. section 4.4) to produce the geospatial clusters. These clusters are used as alternatives to the MCDA technique.
3. The DSS invokes the Facility Location Factors System (c.f. section 4.5) for each identified cluster to gather factors data for that specific location. The retrieved data is used as a criterion by the MCDA technique.
4. The MCDA mechanism initiates and orders the result of the analysis.
5. The DSS returns the result to the decision-maker with a list of ordered suggestions of potential facility locations.

4.3.1 Architecture

The figure 4.5 details the component architecture of the DSS API:

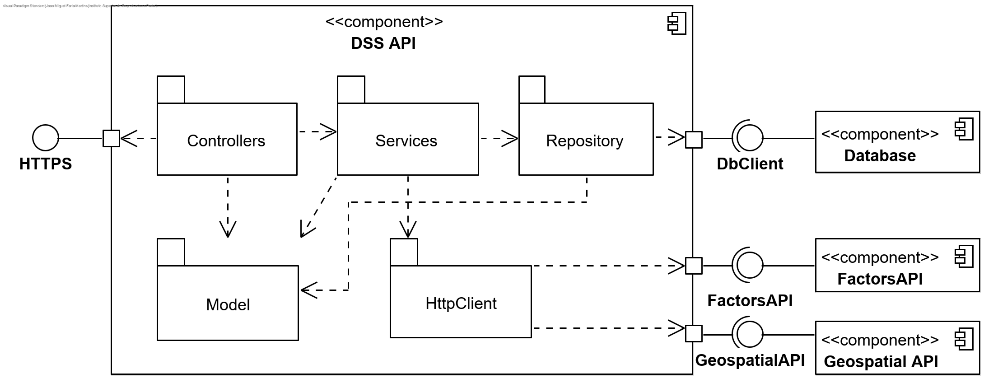


Figure 4.5: Decision Support System - Component diagram

The architecture includes controllers responsible for handling incoming HTTP requests and sending the response back to the DSS client; services to handle the business layer; repositories to handle the persistence layer; HTTP clients to retrieve criterion from the Factors API, and alternatives from the Geospatial API. These criteria and alternatives are returned to the services layer and used on the MCDA service that is responsible for ordering the facility location suggestions.

4.3.2 Domain Model

The domain model (c.f. figure 4.6) has locations and factors that are responsible for generating alternatives and criterion, correspondingly. These alternatives and criterion are used to produce ordered suggestions. Each suggestion can have a set of criterion and is linked to a single alternative.

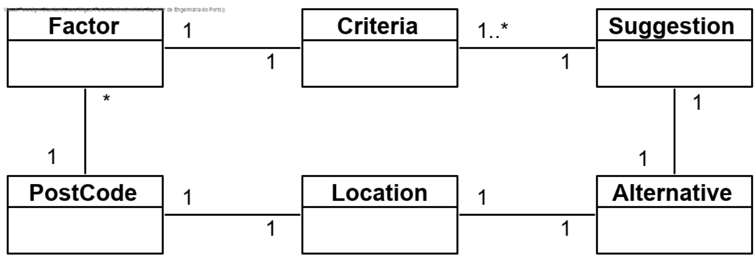


Figure 4.6: Decision Support System - Domain Model

The postcode entity links with the factors and locations and is an essential entity because it is responsible for unifying and converting coordinates into a geographical area. So, the coordinates at the location are used to gather a postcode and the factor data (e.g. crime levels) is relative to an area and not to a single coordinate.

4.3.3 Technologies

To develop the Decision Support System API, it is required to use a programming language to develop a RESTful service to ensure scalability, maintenance and documentation. Another requirement is to have native support of dependency injection (DI) software design pattern used to achieve the Inversion of Control (IoC). The IoC ensures that the solution is low coupled and easy to test.

This system focus essentially in interacting with other API (c.f. section 4.4 and 4.5) through HTTP to gather data to perform the MCDA which is a complex operation. The programming language must support complex programming such as parallel programming to maximise the use of CPU cores by enabling multiple threads to be executed simultaneously.

4.4 Geospatial Analysis System

The Geospatial Analysis System is responsible for analysing the telemetry data and for producing geospatial clusters regarding the potential facility locations. This system has the following steps:

1. The Geospatial Analysis System analyses the existing telemetry regarding the decision-maker business. This analysis results in geospatial clusters as a result of the applied clustering technique.
2. The system returns the result to the DSS to be used as alternatives in the MCDA.

4.4.1 Architecture

The figure 4.7 details the component architecture of the Geospatial Analysis System:

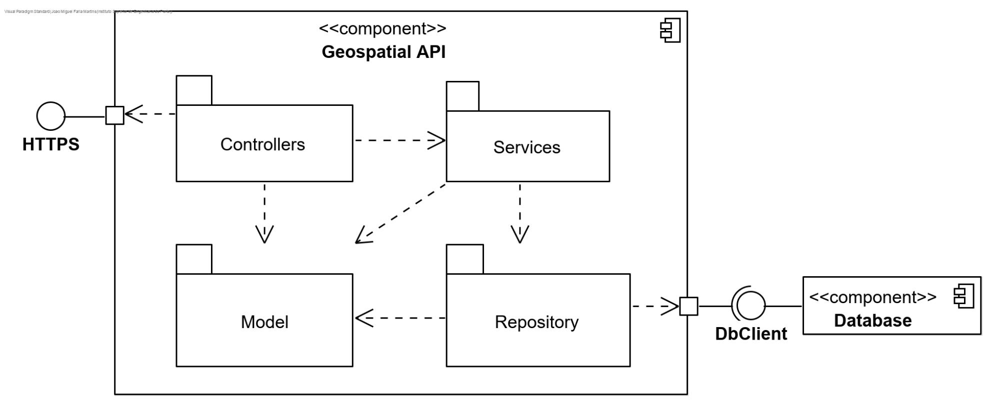


Figure 4.7: Geospatial Analysis System - Component diagram

The architecture is composed by controllers responsible for handle incoming HTTP requests and send the response back with the potential facility locations to the DSS API; services to handle the cluster analysis and return the results to the controller; repository responsible for interacting with the persistence layer.

4.4.2 Domain Model

This domain model (c.f. figure 4.8) has three entities: journey; position; facility location suggestion.

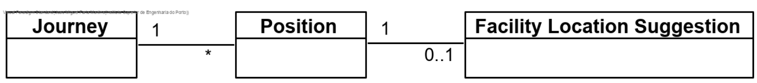


Figure 4.8: Geospatial Analysis System - Domain Model

One journey is composed of a set of positions, and, one of these positions represents the end of the journey. This last position can be or not considered a facility location suggestion according to the clustering algorithm results.

4.4.3 Technologies

To develop the Geospatial Analysis System, it is required to use a programming language to be used to develop the API and the service responsible for applying the geospatial clustering technique (c.f. section 2.5.5) to gather potential facility locations.

The TelemetryDB should be a NoSQL database, and the reasons are the capability to handle:

- Large volume of structured, semi-structured and unstructured data;
- Efficient, flexible and scale-out architecture;
- High availability and horizontal scaling.

4.5 Facility Location Factors System

This system is responsible for retrieving a list of supported factors and retrieve the specific factor's data to be used as a criterion in the MCDA.

This system has the following responsibilities:

- Retrieve a list of supported factors that can be used as the criterion on the MCDA;
- Retrieve specific factor's data for a location.

4.5.1 Architecture

The figure 4.9 details the component architecture of the Factors API.

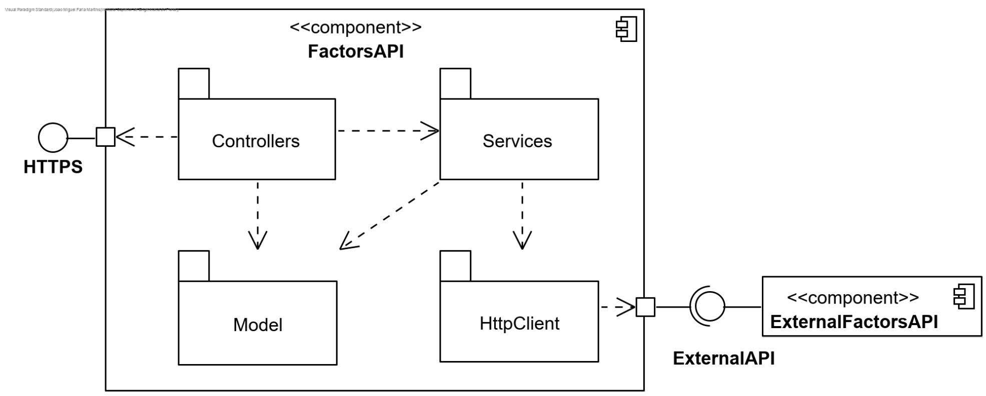


Figure 4.9: Facility Location Factors System - Component diagram

The architecture is composed by controllers responsible for handling incoming HTTP requests and sending the response back to the caller; services to handle responses from the HTTP client module and to return the result to the controller; HTTP client is used to retrieving factor details from external data sources.

4.5.2 Domain Model

This domain model (c.f. figure 4.10) focuses on the factor entity that is composed of a location and by one or more factor data. Six entities inherit from the factor data: build cost, council tax, growth, crime, price per sq. ft. and demographics. This list of factor data is extensible according to the business needs.

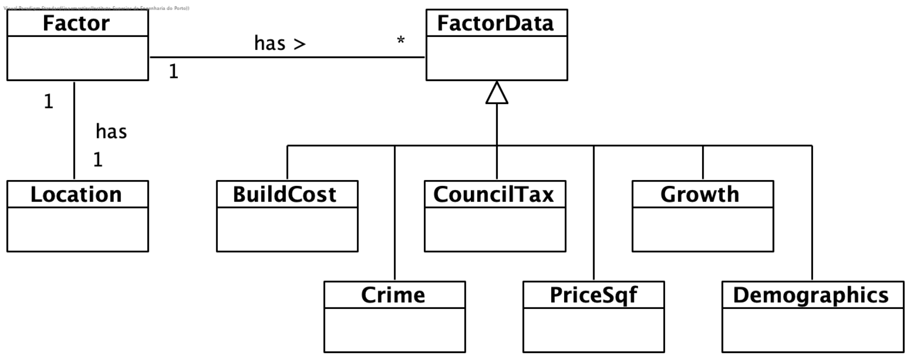


Figure 4.10: Facility Location Factors System - Domain Model

4.5.3 Technologies

To develop the Facility Location Factors System API, it is required to use a programming language to develop a RESTful service to ensure scalability, maintenance and documentation. Another requirement is to have native support of dependency injection (DI) software design pattern used to achieve the Inversion of Control (IoC). The IoC ensures that the solution is low coupled and easy to test.

4.6 Main Flow Execution

This section details in a high level of abstraction the main flow execution of the solution for the generation of facility location suggestions. The sequence diagram (c.f. figure 4.11), details the main interactions between systems.

The decision-maker starts by choosing the factors with more relevance to its business, and these are used to initiate the DSS mechanism. The DSS requests the Geospatial Analysis System to generate potential facility locations, and for each location, retrieve the factors data based on the previous selection made by the decision-maker.

The final responsibility of the DSS is to run the MCDA using the locations as the alternatives and the factor's data as the criterion. The objective of the MCDA is to order the facility location alternatives and retrieve the details for each alternative.

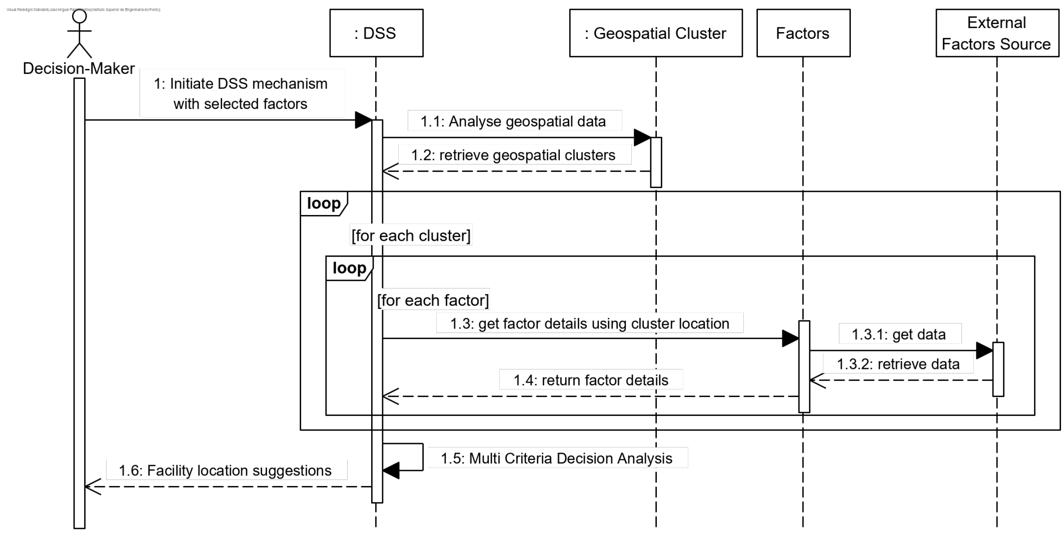


Figure 4.11: System Main Flow Execution - Sequence Diagram

4.7 Summary

In summary, the Design represents an architectural abstraction of the system that describes the system structure, how its elements work together and define guidelines to support future development. The functional requirements and the non-functional requirements (c.f. section 4.1.3) were detailed in this chapter. The functional requirements are:

1. Use Case 1.1: The decision-maker initiates the decision support system mechanism;
2. Use Case 1.2: The decision-maker chooses the facility location;
3. Use Case 1.3: The decision-maker lists the facility location decisions.

Relatively to the system architecture, was developed the component view and the deployment view. Two approaches were proposed on the component view; this project implements the alternative one (c.f. section 4.2.1).

For each system was detailed the internal architecture through a component diagram, and the domain model shows the entities and the interactions between them. Finally, it was created a sequence diagram in order to demonstrate the main flow execution of the solution.

Chapter 5

Implementation

This chapter concerns the function of describing and discussing the implementation of the solution according to the system analysis and design detailed by following good practises of software development. Before reading this chapter, it is advised to read the previous chapter 4 to understand the architecture of the solution as well the developed components for this system. The content of this section is described as follows.

First, the section about the solution overview has the purpose of simplifying the comprehension of the solution. It is used a BPMN diagram to clarify the solution, the development methodology by detailing the Agile[1] software development method and the used tools, and the Continuous Integration/Continuous Delivery mechanism.

Second, each project of the solution is explained by describing their API Resources, following by the Package Diagram that demonstrates the arrangement of the developed project by simplifying the comprehension of the structure and dependencies between elements or modules and the Technology Stack. The MCDA algorithm implemented on the DSS and the clustering algorithm implemented on the Geospatial Analysis System are explained.

Finally, each predefined requirement at section 4.1 is described according to the used approach to fulfil the functional and non-functional specifications.

5.1 Methodology

The development methodology is essential to focus on delivering high-quality working software frequently and consistently. The Agile software development was adopted to guarantee an iterative and incremental development model through the implementation of the Kanban framework. Using this work management system, the team can manage and control the workflow of the project by learning and adapting through the iterative development process.

To implement Kanban was used Trello, a web-based application responsible for holding the control flow by the use of boards and cards that could be classified as the following states: Backlog; Selected for Development; In Progress; Done; Released.

The iterative and incremental model fulfils the developed features followed the Continuous Integration and Continuous Delivery (CI/CD) practises by the use of Git and Bitbucket. Also, it was used Bitbucket Pipelines that automates the process from the code compiling to the building and further deployment of an image by the use of Docker capabilities. In summary, each project has a Dockerfile responsible for assembling an image and a bitbucket

[1] https://agilemanifesto.org/

pipeline script that runs the Dockerfile instructions by building and pushing the image to a registry (c.f. figure 5.1).

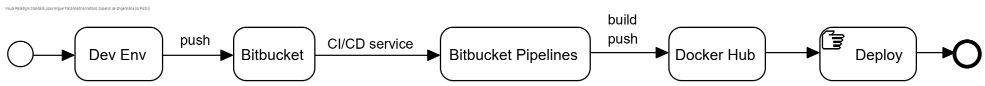


Figure 5.1: CI/CD Pipeline

Finally, to ensure the developed features were aligned with the predefined needs were arranged quality meetings every month with the CTO of the company. Whenever there were questions or guidance needs, meetings were arranged with the supervisors.

5.2 Solution Overview

The solution is composed by three API: Decision Support System API (c.f. section 5.3); Geospatial Analysis System API (c.f. section 5.4); Facility Location Factors System API (c.f. section 5.5). The following BPMN diagram (c.f. figure 5.2) details the solution' overview to simplify the understanding of the solution by representing the high-level components and interactions between developed applications on the generation of facility location suggestions.

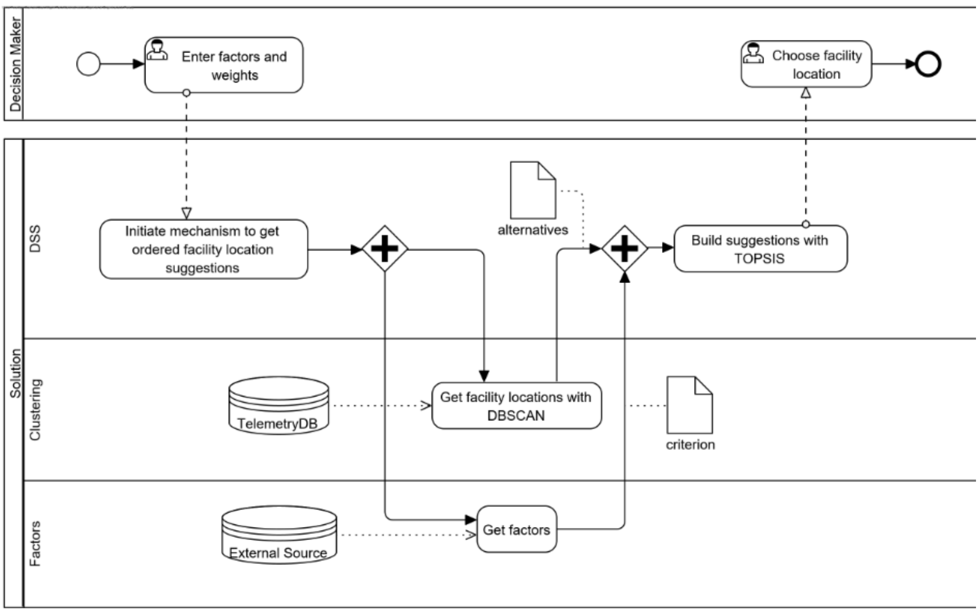


Figure 5.2: Solution Overview - BPMN Diagram

5.3 Decision Support System

The DSS API is responsible for building facility location suggestions and retrieving them to the decision-maker. This system has the following steps:

1. The decision-maker initiates the flow execution by defining the factors that are more relevant to the business;

2. The DSS invokes the Geospatial Analysis System (c.f. section 4.4) to produce the geospatial clusters using the DBSCAN clustering algorithm. These clusters are used as alternatives in the MCDA technique (TOPSIS).

3. The DSS invokes the Facility Location Factors System (c.f. section 4.5) for each identified cluster to gather factors data for that specific location. The retrieved data is used as criterion by TOPSIS.

4. The MCDA mechanism initiates and orders the result of the multi-criteria analysis.

5. The DSS returns the result to the decision-maker with a list of ordered suggestions of potential facility locations.

Finally, the DSS offers the possibility to the decision-makers to decide and persist which suggestions fits better their needs.

5.3.1 API Resources

To fulfil the system requirements, the DSS has two resources (c.f. figure 5.1) and to ensure the user is authorized to access these resources, each HTTP request requires an Authorization Bearer Token. The decisions resource is responsible for retrieve and defines the facility location decision.

The suggestions resource is the endpoint responsible for returning the ordered facility location suggestions. The query string used on the suggestions resource is a base64 query due to the complexity of the request. This query is the result of an encoding of a JSON structure composed by account identifier, factor identifiers, factor weights (the total sum of each weight must be equal to 1) and classification of the factors as cost or benefit (0 and 1 respectively).

Table 5.1: Decision Support System - API Resources

URL	**Methods**	**Query String**
`/decisions`	GET; POST	
`/suggestions`	GET	query={encoded-base64-query}

The JSON response of the suggestions endpoint is detailed on the listing 5.1 and gives the decision-maker an array of suggestions with the location details, the selected factors, the values for each factor and the performance score that is used to order the facility location suggestions.

```
[
    {
        "id": {string},
        "timestamp": {DateTime},
        "location": {
            "postCode": {string},
            "outCode": {string},
            "inCode": {string},
            "latitude": {double},
            "longitude": {double},
            "clusterSize": {integer}
        },
        "factors": {string[]},
        "factorValuesByLocation": {double[]},
        "performanceScore": {double}
    }
]
```

Listing 5.1: Decision Support System - HTTP JSON Structure Response of /suggestions

5.3.2 Package Diagram

Three main components compose the DSS package diagram (c.f. figure 5.3): Controllers package used as the entry point of the API and a gateway between the received HTTP request and the domain logic; Services package responsible for handling the business logic and provides an abstraction to the persistence layer; Repositories package used as a gateway between the domain/business layer and the data access layer, which is the layer responsible for accessing the database and applying database operations.

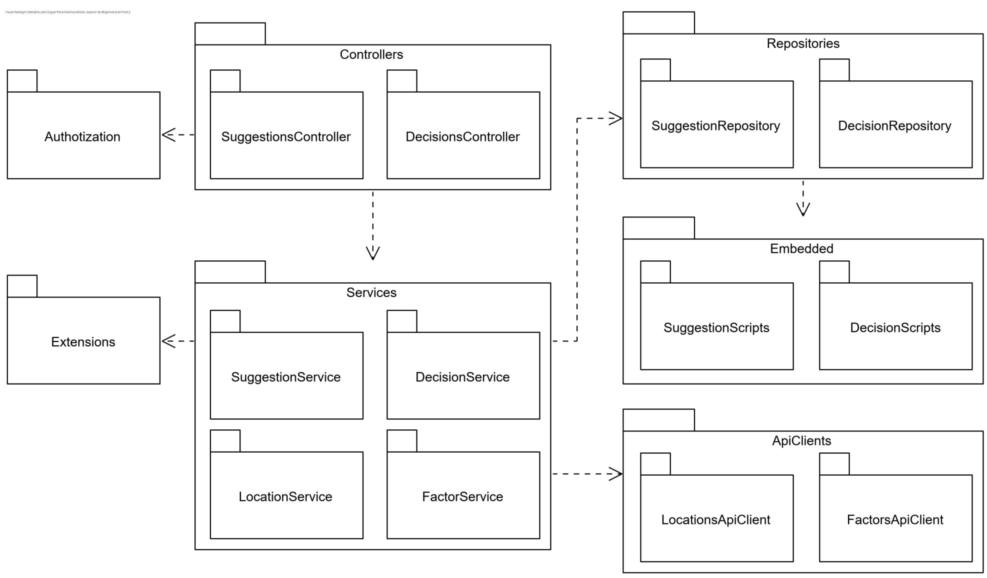


Figure 5.3: Decision Support System - Package diagram

5.3.3 Multi-Criteria Decision Analysis Algorithm

To calculate the performance scores according to the provided alternatives (locations) and the criterion (factors) was implemented the TOPSIS[2] that is the chosen MCDA method. The steps of the TOPSIS implementation are detailed at the algorithm 5.1 and follows the steps described in section 2.4.2.

The TOPSIS algorithm requires as input parameters a matrix with the alternatives and the values for the factors, the factor weights and the classification (cost or benefit) for each selected factor.

Algorithm 5.1 Decision Support System - TOPSIS Algorithm

1: Set normalized matrix
2: Set normalized weighted matrix
3: Set positive and negative ideal solutions
4: Set relative closeness coefficient to ideal solution

This implementation handles factors two different criteria classifications (c.f. table 2.3): cost and benefit. The algorithm considers the negative ideal solution when the criteria are classified as cost (e.g. council tax band or crime rate) and the positive ideal solution when the criteria are classified as a benefit (e.g. growth zones or education level). Finally, the algorithm returns an array with the performance scores for each alternative that is used to rank the suggestions.

5.3.4 Technology Stack

To develop the Decision Support System API (c.f. table 5.2), it is used .NET Core framework using C# (version 7.3) as the programming language. This technology is used to develop a RESTful service to ensure scalability, maintenance and documentation. Another reason to use this framework is the native support of dependency injection (DI) software design pattern used to achieve the Inversion of Control (IoC).

Table 5.2: Decision Support API - Technology Stack

Technology	**Version**
.NET Core	2.2
PostgreSQL	12

This system focus essentially in interacting with other API (c.f. section 4.4 and 4.5) through HTTP to gather data to perform the MCDA which is a complex operation.

The database used in the decision support system is a SQL database. The PostgreSQL 12 was used in this project because it is robust, open-source object-relational database system with reliability, feature robustness, and performance (*PostgreSQL* 2020).

[2]The TOPSIS implementation was based on a project found on a GitHub repository: https://github.com/muratkuru/TOPSIS

5.4 Geospatial Analysis System

The Geospatial Analysis System is responsible for analysing the telemetry data and for producing geospatial clusters regarding the potential facility locations. This system has the following steps:

1. The Geospatial Analysis System analyses the existing telemetry regarding the decision-maker business. This analysis return geospatial clusters as a result of the applied DBSCAN clustering technique.
2. The API returns the identified clusters to the DSS to be used as alternatives in the MCDA.

5.4.1 API Resources

The Geospatial Analysis System has one resource (c.f. figure 5.3) and to ensure the user is authorized to access, each HTTP request requires an Authorization Bearer Token. The cluster's resource is responsible for returning the four most significant clusters calculated by DBSCAN Algorithm.

Table 5.3: Geospatial Analysis System - API Resources

URL	Methods	Query String
`/clusters`	GET	account={client-account-id}

The JSON response of the clusters endpoint is detailed on the listing 5.2 that returns an array of clusters with the latitude, longitude and number of data points of those clusters. The response shows the number of clusters, but there are only considered the four most relevant clusters to be used as alternatives by the DSS.

```json
{
    "clusters": [
        {
            "latitude": {double},
            "longitude": {double},
            "size": {integer}
        }
    ],
    "number_clusters": {integer}
}
```

Listing 5.2: Geospatial Analysis System - HTTP JSON Response of /clusters

5.4.2 Package Diagram

Three main components compose the Geospatial Analysis System package diagram (c.f. figure 5.4): Controllers package used as the entry point of the API and a gateway between the received HTTP request and the domain logic; Services package responsible for handling the business logic and provides an abstraction to the persistence layer; Repositories package used as a gateway between the domain/business layer and the data access layer, which is the layer responsible for accessing the database and applying database operations.

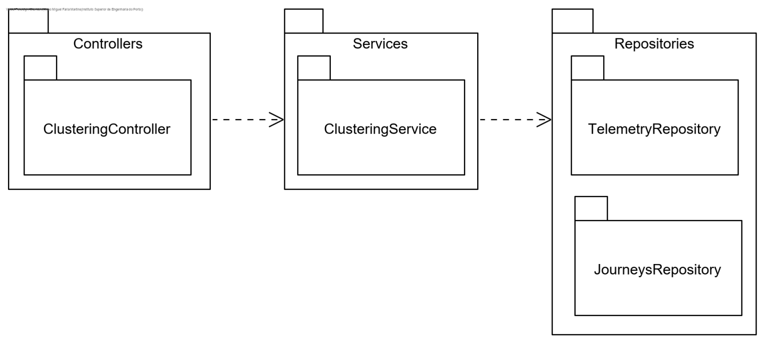


Figure 5.4: Geospatial Analysis System - Package diagram

5.4.3 Data Handling

This subsection focus on the steps to analyse and clean the data used on this project. The data is explained, cleaned and detailed how it was used to fit this solution. The solution is prepared to handle any source of data as long as the data complains about the predefined rules. These steps are always followed with examples to simplify the reading comprehension.

This project was developed on behalf of Fonix Telematics, and the data used in this project is acquired from the internal fleet management system. Any external source of data can be used if the data follows the data structure rules. It is important to note that the system can only produce reliable suggestions if the input data has journeys from at least the last six months.

Data Set Structure

The data set is the data used by the clustering algorithm to produce location suggestions based on the existing data points. The Geospatial Analysis System handles two separate data collections persisted on a MongoDB instance: (1) journeys data set and (2) telemetry data set. Furthermore, the journeys used by the clustering algorithm are extracted from the intersection of the *EndGpsTimestamp* from the journeys collection and the *GpsTimestamp* from the telemetry collection. This intersection results on a set of coordinates to be used by the clustering algorithm.

The table 5.4 details the journeys data structure. This structure is regarding the summary of the completed journeys and, with this data, it is possible to identify starting and ending locations, total mileage and fuel cost for each vehicle. The geospatial system is only interested in the *EndGpsTimestamp*.

Table 5.4: Journeys data structure

Key	Description	Relevance
AccountId	Company Unique Identification	Irrelevant
AssetId	Asset Unique Identification	Irrelevant
DataPoints	Total number of positions	Irrelevant
EndGpsTimestamp	**End time of journey**	**Relevant**
EndAddress	End journey address	Irrelevant
FuelUsed	Fuel used in the journey	Irrelevant
FuelCost	Fuel cost of the journey	Irrelevant
Mileage	Journey mileage (km)	Irrelevant
StartGpsTimestamp	Start time of journey	Irrelevant
StartAddress	Start journey address	Irrelevant

The telemetry data structure is detailed in the table 5.5 and its regarding to each transmitted data point. One journey has n data points. The geospatial system is interested on the *GpsTimestamp*, *Latitude* and *Longitude*.

Table 5.5: Telemetry data structure

Key	Description	Relevance
AccountId	Company Unique Identification	Irrelevant
Altitude	Location Altitude	Irrelevant
AssetId	Asset Unique Identification	Irrelevant
GpsTimestamp	**Transmitted data timestamp**	**Relevant**
Heading	Vehicle direction on degrees	Irrelevant
Ignition	Ignition status	Irrelevant
Latitude	**Latitude**	**Relevant**
Longitude	**Longitude**	**Relevant**
Speed	Speed	Irrelevant

Data Analysis

The journeys used by the clustering algorithm are extracted from the intersection of the EndGpsTimestamp from the journeys collection and GpsTimestamp from the telemetry collection resulting on a set of coordinates to be used by the DBSCAN algorithm (further details on section 5.4.4). By a domain decision, the clustering results are ordered by size, and there are only considered the four most relevant clusters to the solution. The minor clustering found by the DBSCAN algorithm is considered as irrelevant due to an insufficient number of journeys to be considered a potential facility location.

To verify the solution's approach was used data from two distinct companies, these are called as Company A and Company B from now on. The table 5.6 holds the details about the number of rows used by the solution to test and execute it.

Table 5.6: Data set size details

Identifier	No. Journeys	No. Telemetry
Company A	161.220	8.168.429
Company B	140.535	5.964.332

In order to simplify the comprehension of the developed algorithm the following figures help to visualize the data intersection between the journeys and telemetry (c.f. figures 5.5 and 5.7), and the result of the clustering algorithm (c.f. figures 5.6 and 5.8) considering the geospatial locations on a map of the four most relevant results from the clustering algorithm.

Data Visualization - Company A

Company A has a robust network around London city resulting on 161.220 journeys and 8.168.429 positions on six months.

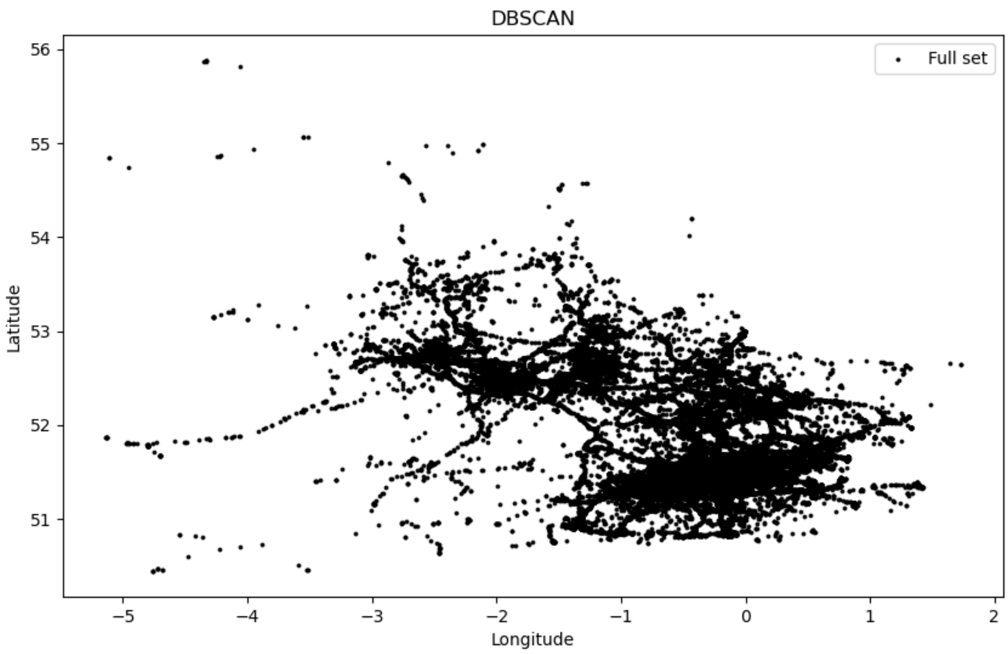


Figure 5.5: Clustering algorithm execution - Company A

By visually analysing the data distribution on the figure 5.5 the data points are centred on London, and the clustering algorithm result (c.f. figure 5.6) shows that were found the three most significant clusters on that area. Finally, the last cluster is located around Wolverhampton and has more than 6000 journeys on the last six months.

Figure 5.6: Result of clustering algorithm execution - Company A

Data Visualization - Company B

Company B has a distributed network all over the United Kingdom, resulting in 140.535 journeys and 5.964.332 positions on six months.

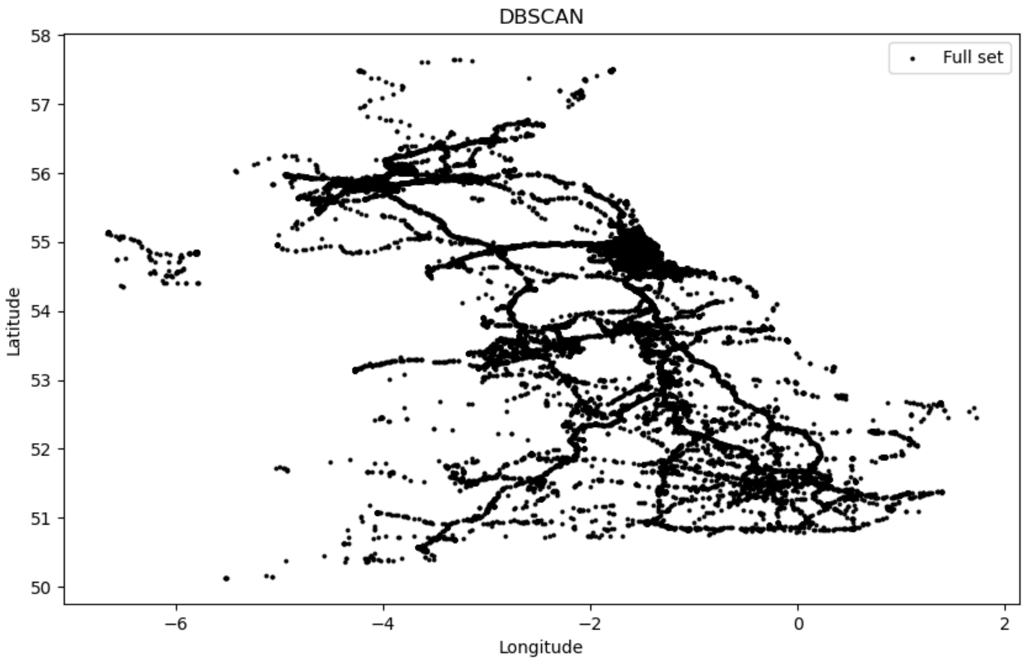


Figure 5.7: Clustering algorithm execution - Company B

By analysing the data distribution on the figure 5.7 the data points are all over the United Kingdom, and the clustering algorithm result (c.f. figure 5.8) shows that the most relevant clusters are on Newcastle. Finally, the last cluster is located around Glasgow.

Figure 5.8: Result of clustering algorithm execution - Company B

5.4.4 Clustering Algorithm

As defined in section 2.5, the library used to apply the DBSCAN algorithm to the geospatial data is the scikit-learn. The function invocation requires four parameters: epsilon, minimum samples, algorithm and metric, applying these parameters to the coordinates converted to radians. The haversine metric and ball tree algorithm are used as parameters to calculate great-circle distances between coordinates. The value selection for the epsilon and minimum samples are explained in section 5.4.4.

```
db = DBSCAN(eps=epsilon, min_samples=min_samples, algorithm='ball_tree',
    metric='haversine').fit(np.radians(coordinates))
```

Listing 5.3: Scikit-learn DBSCAN

The haversine distance is the angular distance between two points on the surface of a sphere. As the Earth is nearly spherical, the haversine formula provides a good approximation of the distance between two points, with less than 1% error (*Scikit-Learn Haversine* 2020). The ball tree algorithm is used for the Nearest Neighbour search and was the technique with better performance results comparing to the other available algorithms: KD Tree and brute force. The brute force is theoretically the most accurate method due to the consideration of all data points. However, this method has speed and efficiency issues due to the high number of data points.

Determining the Parameters Eps and MinPts

The parameters of the DBSCAN algorithm are the *Eps* and the *MinPts*. The *Eps* value is the physical distance from each location, and the *MinPts* is the minimum cluster size. A more considerable value for *Eps* results in broader clusters, while a smaller value produces smaller narrower clusters. The more considerable value of *MinPts* produces a more significant cluster but may exclude smaller areas as it attempts to merge them in one large cluster. Ideally, before development, we would have to know the values of these parameters for each cluster, but there is no easy way to get this information in advance for all clusters on the database (Ester et al. 1996).

To develop an approach for this problem was decided that the way to define these parameters is based on the domain knowledge (c.f. section 6.3.2). The *Eps* was defined according to an acceptable distance between points to be considered part of a cluster.

The *Eps* value and the coordinates are converted to radians because scikit-learn's haversine needs radian units (Boeing 2018). The *Eps* value was set as 1.5 kilometres defining the maximum distance that points can be from each other to be considered a cluster (Boeing 2018). As stated to use the haversine distance, it is required to convert the distance in kilometres to radians as follows:

$$eps = \frac{distance}{earthRadius}$$

To set the *MinPts* was defined as the adequate number of visits to an area to be considered a cluster. In this case, running the DBSCAN algorithm with a high value for *MinPts* is more likely to produce clusters with more relevance identifying those that are more significant clusters. This value was set to 2500, meaning a core point can only be considered a potential facility location if the number of journeys to that area is more significant than the *MinPts*.

This system analyses data from the last six months, meaning that 2500 journeys divided by six months are approximately 416 journeys per month. The 416 journeys per month divided by 22 business days is approximately 19 journeys per day that are located maximum 1.5 km from the core point. Minimizing the number of relevant clusters by maximizing the minimum points surrounding a core point is crucial to simplify the interpretation and identity of the results of the most relevant clusters. In summary, the *Eps* value is 1.5 and the *MinPts* is 2500 (c.f. section 6.4.2).

5.4.5 Technology Stack

To develop the Geospatial Analysis System (c.f. table 5.7), it is used Python (version 3.8.3). This programming language is used to develop the API and the service responsible for applying the geospatial clustering technique (c.f. section 2.5.5) to gather potential facility locations.

Table 5.7: Geospatial Analysis System - Technology Stack

Technology	Version
Python	3.8.3
MongoDB	4.2.6

The TelemetryDB is a NoSQL database (MongoDB - version 4.2.6), and the advantages are the capacity to handle:

- Large volume of structured, semi-structured and unstructured data;
- Efficient, flexible and scale-out architecture.

The MongoDB replaces the tables by collections, the rows by documents and columns by fields. The main advantages of using MongoDB in this project are the flexible data storing using JSON-like documents and the distributed database to ensure the high availability and horizontal scaling (*MongoDB* 2020).

5.5 Facility Location Factors System

This system is responsible for retrieving a list of supported factors and retrieve the specific factor's data to be used as a criterion in the MCDA.

This system has the following responsibilities:

- Retrieve a list of supported factors (c.f. table 5.8) that can be used as the criterion on the MCDA;
- Retrieve specific factor's data for a location.

Table 5.8: List of supported facility location factors

Factor	Description	Details/Metrics
build-cost	Construction Costs	Total cost of construction
cluster-size	Cluster Size	Number of cluster points
council-tax	Council Tax Band	Value evaluation of the property
crime	Safety levels	Crime rank per area
demographics	Indices of Deprivation	Employment Rank, Education and Skills, Health and Living environment
growth	Growth zones	Location evolution and investment
prices-per-sqf	Prices per Square Foot	Price per Square Foot

5.5.1 API Resources

The Facility Location Factors System has one resource (c.f. figure 5.9) and to ensure the user is authorized to access these resources, HTTP requests require an Authorization Bearer Token. The factors resource is responsible for retrieving a list of supported factors (c.f. table 5.8) and retrieve the details by id for a specific factor according to the postcode.

Table 5.9: Facility Location Factors System - API Resources

URL	Methods	Query String
`/factors`	GET	
`/factors/{id}`	GET	postCode={post-code}

It is important to note that the application has only one resource for N factors because the API can resolve the requested factor by identifier. Meaning the system that integrates the Facility Location Factors System does not need to do changes to the application if new factors are implemented.

```
{
    "id" : {string},
    "label" : {string},
    "data": [
        {
            "label": {string},
            "value": {object}
        }
    ]
}
```

Listing 5.4: Facility Location Factors System - HTTP JSON Response of /factors/{id}

The JSON response of the factors by id is described on the listing 5.4 that returns an object with the data to be used as criterion on the DSS.

5.5.2 Package Diagram

Three main components compose the Facility Location Factors System package diagram (c.f. figure 5.9): Controllers package used as the entry point of the API and a gateway between the received HTTP request and the domain logic; Dependency Injection Resolver responsible for instantiating and inject the available factor queries; Factor Query is the supporting factor that is used for gathering the data from an external source, and each factor query is identified with a unique Attribute.

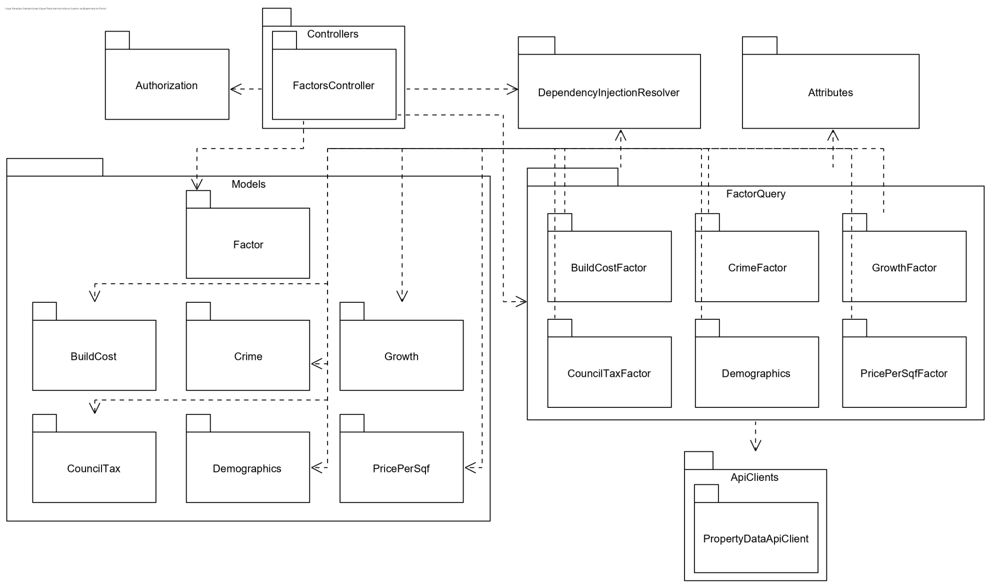


Figure 5.9: Facility Location Factors System - Package diagram

5.5.3 Factor Implementation

This section has the purpose of detailing and explaining the mechanism to handle various factors in a generic approach. Each factor has a attribute that identifies the factor with a custom definition (c.f. listing 5.5).

```
[Factor( "council-tax" )]
public class CouncilTaxFactor : IFactorQuery
```

Listing 5.5: Facility Location Factors System - Factors Attribute

Every factor implements the method from the factor query interface that has the responsibility to gather the specific factor data from an external API and to return a generic factor object transformed according to listing 5.4.

The factors are injected through dependency injection and identified by their custom definitions. When the factors resource receives an HTTP request, the DI resolver tries to find the requested factor and executes it. This approach allows the system to have a single resource, simplifying the implementation of new factors and minimizing additional changes on the API that integrates this system.

5.5.4 Technology Stack

To develop the Facility Location Factors System API (c.f. table 5.10), it is used .NET Core framework using C# (version 7.3) as the programming language.

Table 5.10: Facility Location Factors System - Technology Stack

Technology	Version
.NET Core	2.2

This technology is used to develop a RESTful service to ensure scalability, maintenance and documentation. Another reason to use this framework is the native support of DI software design pattern used to achieve the IoC. The IoC ensures that the solution is low coupled and easy to test.

5.6 Requirements

The developed requirements are described at section 4.1. The implementation of each use case considers the predefined functional and non-functional requirements to fulfil the solution specifications.

5.6.1 Use Case 1.1: Initiate the decision support system mechanism

The authenticated and authorised decision-maker initiates the DSS mechanism by defining the factors (c.f. table 5.8) that are more relevant to the facility location selection. The system receives the chosen factors and triggers the facility location suggestions process (c.f. algorithm 5.2).

Algorithm 5.2 Decision Support System - Pseudocode - MCDA algorithm

1: Get facility location suggestions - Alternatives (c.f. algorithm 5.3)
2: Get factors - Criterion (c.f. algorithm 5.4)
3: Execute TOPSIS method using gathered alternatives and criterion - MCDA
4: Build and return ordered facility location suggestions to the decision-maker

To get alternatives to be used by the MCDA on the DSS, the algorithm 5.3 describes the implemented steps to identify the facility location suggestions. Lastly, the Geospatial Analysis System returns the HTTP JSON response (c.f. listing 5.2) to the DSS.

Algorithm 5.3 Geospatial Analysis System - Pseudocode - Build Facility Location Suggestions

1: Initiate clustering process request
2: Establish connection with MongoDB client
3: Get journeys and telemetry data
4: Get last position from each journey using telemetry data (e.g. figures 5.5 and 5.7)
5: Apply DBSCAN clustering algorithm
6: Build facility location suggestions
7: Retrieve facility location suggestions (e.g. figures 5.6 and 5.8)

To get the criterion to be used by the MCDA on the DSS, the algorithm 5.4 describes the implemented steps to get the factor data based on a postcode. This Facility Location Factors System is prepared to implement new factors in a practical manner, the system gets the data for a specific factor and builds a generic factor that is used as HTTP JSON response (c.f. listing 5.4).

Algorithm 5.4 Facility Location Factors System - Pseudocode - Get factor details by id

1: Get factor query resolver by id (c.f. figure 5.8)
2: Get factor details (c.f. figure 5.8) by factor criteria (default by post code) from external API
3: Build generic factor entity from specific factor
4: Return result

Note that if necessary to implement new factors it is only required to implement the specific factor because the Facility Location Factors System is capable of transforming it to a generic structure and the DSS is receiving this structure that is the same for all factors. The section 6.4.4 details a real execution scenario of this use case.

5.6.2 Use Case 1.2: Choose facility location

The authenticated and authorised decision-maker makes its decision after getting and analysing the produced facility location suggestions (c.f. algorithm 5.5).

Algorithm 5.5 Decision Support System - Pseudocode - Create facility location decision

1: Choose facility location suggestion
2: Build decision from selected suggestion
3: Persist facility location decision on relational database

The decision-maker sends an HTTP request with the preferred facility location only by sending the suggestion identifier, the controller invokes the service layer that is responsible for invoking the repository layer. Finally, the decision is persisted on the database. It is logical the decision-maker necessity to view their decisions, so it was implemented the use case responsible for retrieving the decisions (c.f. section 5.6.3).

5.6.3 Use Case 1.3: List facility location decisions

The authenticated and authorised decision-maker lists its previous facility location decisions (c.f. algorithm 5.6).

Algorithm 5.6 Decision Support System - Pseudocode - List facility location decisions

```
1: Request for previous facility location decisions
2: Retrieve list of facility location decisions from relational database
```

The decision-maker sends an HTTP request for listing the existing facility location decisions. The controller invokes the service layer that is responsible for invoking the repository layer. Finally, the previous decisions are retrieved from the database.

5.7 Summary

In summary, this solution was developed using good software development practices. The solution is composed of three main projects: (1) Decision Support System; (2) Geospatial Analysis System; (3) Facility Location Factors System.

The Decision Support System is an API responsible for building facility location suggestions through an MCDA and retrieving them to the decision-maker. This project was implemented using .NET Core as a coding framework and PostgreSQL as a persistence layer.

The Geospatial Analysis System is an API responsible for analysing the telemetry data and for producing geospatial clusters regarding the potential facility locations to be used as alternatives by the MCDA. This project was implemented using Python as a coding language and MongoDB as a persistence layer.

The Facility Location Factors System is an API responsible for retrieving a list of supported factors and retrieve the specific factor's data to be used as a criterion by the MCDA. This project was implemented using .NET Core as a coding framework.

Chapter 6

Evaluation

The Evaluation chapter concerns the function of describing the process of evaluation and analysis of the solution. The first step is regarding the hypothesis identification, then a description of the evaluation indicators, the explanation of the evaluation methodology and, finally, the evaluation of experiments and results.

6.1 Hypothesis

A research hypothesis is a quantifiable, verifiable and testable predictive statement about the possible outcome of scientific research on a particular property of a population. Specifying the research hypothesis is one of the essential steps in planning a scientific quantitative research study (Lavrakas 2008).

The main goal of this project is to provide suggestions for facility locations through the telemetry analysis. The hypothesis is focused on evaluating the DSS effectiveness and the quality of the suggested locations.

The developed decision support system (DSS) has three main components: (1) identification of potential locations based on the analysis of their telemetry coming from their assets, (2) analysis of the factors that may affect the facility location choice, and (3) provide a set of suggestions based on the multi-criteria analysis.

To verify and validate the effectiveness of a DSS, it is fundamental to develop a system evaluation. The evaluation allows the developers to understand how well the system works and how it addresses the need of its users (Shafinah et al. 2010). The evaluation of the hypothesis focuses on the quality of the DSS effectiveness results, clustering analysis and software testing analysis.

To evaluate the dissertation work was defined as the following hypothesis:

- The system is capable of suggesting strategic, accurate and valuable facility locations based on fleet analysis and business needs.

6.2 Evaluation Indicators

The decision support system evaluation analyses the quantitative data from the respondents of the survey. The evaluation indicators examine the given data to formulate and classify the questions developed in the survey (Shafinah et al. 2010). The table 6.1 has the chosen factors used to evaluate the DSS model through the inquiry method.

Table 6.1: Usability factors (Shafinah et al. 2010)

Factor	**Definition**
Efficiency	System efficiency time to use
Understandability	The capability of the system to enable the user to understand the system and its features and conditions of use
Accuracy	The correctness of the output information
Effectiveness	The capability of the system to achieve the specified goals

The clustering evaluation is made by verifying the quality of the DBSCAN parameters according to the output and by analysing metrics provided by the result of the clustering algorithm. As a constraint, the used data sets must be referent to companies with at least fifty assets with daily driving for at least the past six months, to identify the most common journeys to produce suggestions of the potential facility locations.

Finally, the system is tested using unit and load tests. The unit tests are applied to the whole system to verify each unit of the solution. The performance under load is measured by using metrics provided by the result of the load tests execution. The metrics analysis focus on response times, average response time, user load and total requests.

6.3 Evaluation Methodology

As described on the Hypothesis, the evaluation of the hypothesis focuses on three aspects: (1) the evaluation of quality of the decision support system by measuring the efficiency, understandability, accuracy and effectiveness, (2) the clustering analysis, and (3) the system test analysis.

6.3.1 Decision Support System Quality

According to Shafinah et al. 2010, one of the used approaches to evaluate the DSS is system analysis. The characteristics that define this type of testing are: (1) the goal is to improve a product, (2) the participants are real users, (3) the users do real tests, (4) the tester obverse and record the participants and (5) the participants suggest changes and identify problems in the solution (Dumas and Redish 1993).

Three main evaluation techniques have been used to measure key factors: inspection methods, inquiry methods and testing methods. This project focuses on inquiry methods by gathering the user's opinions and preferences of the system characteristics (Shafinah et al. 2010). This method requires quantitative data, which is obtained by users' opinions from the responses to the developed survey (c.f appendix A).

The chosen evaluation methodology has the following steps: (1) identification of the problem statement, (2) definition of the concept of quality for this problem, (3) choosing of the evaluation method, (4) determining the evaluation measurement factors (c.f. table 6.1), (5) formulation of the survey questions and finally, (6) data collection and analysis.

The survey is divided into two parts to analyse the perceived usefulness and the perceived ease of use. The survey uses the Likert scale (c.f. table 6.2).

Table 6.2: Likert scale

Scale	Description
1	Strongly disagree
2	Disagree
3	Neutral
4	Agree
5	Strongly agree

In overview, the survey focuses on the following items to evaluate the quality of the system:

- The system is clear and understandable;
- The system has correct principles and assumptions;
- The system achieved the stated objectives;
- The model has captured the main real-world processes and limitations;
- The system works efficiently;
- This system adds value to the business strategy;
- Rate the overall satisfaction.

Before data collection and analysis, it is necessary to define the number of respondents. An empirical study made by Faulkner in 2003 shows that increasing the number of users, the results are more reliable when compared to the five users defined by Nielsen in 1993. Even so, this project uses five users to respond to the survey due to limitations.

Finally, to analyse the inquiries, the quality of the solution will be calculated through the survey analysis by interpreting the responses.

6.3.2 Clustering Analysis

The clustering technique is used to extract hidden information pattern from a large dataset which is useful in decision making (Babur et al. 2015). The purpose of the clustering is to divide data into groups containing similar data, which in this project is to identify the most significant facility location suggestions.

Clustering analysis is a challenging topic and as stated by Jain and Dubes 1988 more than 30 years ago: "without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage", that seems to be applied nowadays.

Clustering validation is one of the most challenging steps, which is the objective and quantitative assessment of clustering results. There are numerous relative validity criteria to verify globular clusters (e.g. Silhouette coefficient), but not all data are composed of globular clusters (Moulavi et al. 2014).

Density-based clustering algorithms search for high-density areas of points that are not necessarily globular, which is the case of the majority of the data that are analysed to identify potential facility locations. Each company has their journeys and patterns that can have high-density areas separated by low-density areas, possibly containing noise objects that can result on failure when applied relative validity indices proposed for globular cluster validation (Moulavi et al. 2014).

The following subsections describe a few clustering analysis approaches that were considered for this project. For each clustering analysis technique, are considered the following steps:

1. Select fleet companies with at least 50 assets with daily driving;
2. Select fleet companies using the system at least for 6 months;
3. Apply the clustering algorithm to geospatial data;
4. Apply clustering analysis technique;
5. Analyse and process clustering results - execution details, output and metrics.

The DBSCAN minimum points was set by the execution of experiments and analysing the results (c.f. section 6.3.2).

Determining DBSCAN Minimum Points

The DBSCAN clustering algorithm has two parameters: *Eps* and *MinPts*. The *Eps* value is the physical distance from each location, and the *MinPts* is the minimum cluster size (Ester et al. 1996). The *MinPts* parameter sets the minimum points within the distance defined by the *Eps* value. This means that two points are considered to be neighbors if the distance between them is less or equal than *Eps* (Ester et al. 1996).

There is an effective heuristic to determine *Eps* and *MinPts* parameters applied to the smallest cluster. Therefore, the same values are used for all clusters. The density parameters of the smallest cluster are good candidates for these parameters specifying the lowest density, which is not considered to be noise (Ester et al. 1996).

Therefore, defining the *Eps* and *MinPts* values according to the empirical study (Ester et al. 1996) produces small-sized clusters and don't capture the most significant clusters. This heuristic (proposed by Ester et al. 1996) is not used on this project since it wants to capture the most significant clusters.

These parameters were defined based on experiments and domain knowledge (c.f. 6.4.2) applied to data sets from the Company A and Company B. The *Eps* was defined according to an acceptable distance between points to be considered part of a cluster. The *MinPts* was defined by finding the adequate number of visits to an area to be considered a potential facility location.

More details in depth about the selection of the *Eps* and the *MinPts* values can be found on the section 5.4.4.

Density-Based Clustering Validation

A possibility to evaluate this system is a validation index for density-based, arbitrarily shaped clusters named as Density-Based Clustering Validation (DBCV) that was proposed by Moulavi et al. 2014. This method evaluates the clustering quality based on the relative density connection between pairs of objects. The index is used to compute the density of objects and to evaluate the within and between cluster density connectedness of clustering results (Moulavi et al. 2014).

Other Measures

As stated on 6.3.2, a method to verify the globular cluster is the Silhouette coefficient, that provides a score between -1 and 1 to verify the quality of clustering. This method measures how well an object is classified in its cluster by interpreting and validating the consistency within clusters. This works well on globular clusters but can fail on non-globular clusters representing a drawback to applying this approach to this project. Therefore, this coefficient was not used to analyse the clustering results.

There are other measures available to evaluate the clustering results. However, since this is unsupervised learning, and there is no correct solution to arbitrary data, these cannot be used in this problem. The methods that require a parameter with the expected output to compare with the algorithm execution are Homogeneity; Completeness; V-measure; Adjusted Rand Index; Adjusted Mutual Information.

6.3.3 Software Testing

Software testing determines the quality of software by improving security, product quality and customer satisfaction. To deliver quality software, proper testing is required. Testing procedures reduces maintenance costs, and the results are more reliable and accurate. In this project were developed the following tests: (1) unit testing; (2) load testing.

Unit Testing

Unit testing evaluates individual components and functions of a system and ensures it performs as expected by verifying the accuracy of each unit. These tests help with maintaining and changing the code, reduces defects and bugs.

The unit tests were applied to the three projects: Decision Support System API, Geospatial Analsyis System API and Facility Location Factors System API.

The tool to be used to develop these tests is the xUnit.net that is open source, community-focused unit testing tool for the .NET Core Framework.

Load Testing

To evaluate the performance of the system under real-life based load conditions, the functionality responsible for generating geospatial suggestions to the customer was tested to ensure that the application meets speed and stability under expected workloads.

The used loading testing tool was the JMeter that is used to create and perform these tests. To create a test plan, it is necessary to configure a few parameters, such as:

- HTTP endpoint request;
- Number of Threads - total number of virtual users that are making HTTP requests;
- Ramp-up Period (in seconds) - how long it takes to reach the full number of threads;

- Loop Count - number of requests made by each thread.

6.4 Experiments and Results

This section holds the Decision Support System Quality evaluation through the analysis of the gathered survey results, the Clustering Analysis experiments by detailing the definition process of the DBSCAN parameters, the software testing to ensure the system performs as expected and the software experiment that describes the stages that the solution has when producing facility location suggestions.

6.4.1 Decision Support System Quality

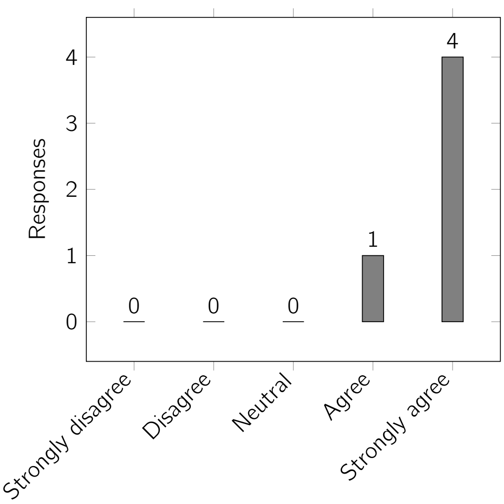
The quality of the DSS was evaluated through a survey (c.f. appendix A) that was used to gather opinions about the developed solution that were answered by five users. The survey was divided into four evaluation measurement factors: (1) Efficiency, (2) Understandability, (3) Accuracy, and (4) Effectiveness.

Efficiency

The efficiency measurement tries to capture the system efficiency time to use. Were defined two questions about this subject:

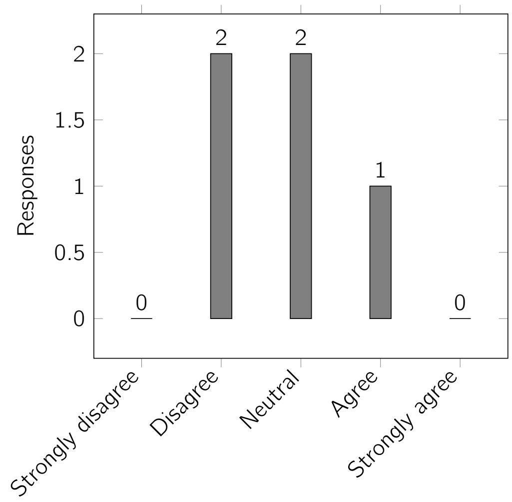
1. Help to choose appropriate facility location more quickly (c.f. figure 6.1);

Figure 6.1: Efficiency - Help to choose appropriate facility location in a sustained manner



2. Response time of the system is acceptable (c.f. figure 6.2).

Figure 6.2: Efficiency - Response time of the system is acceptable



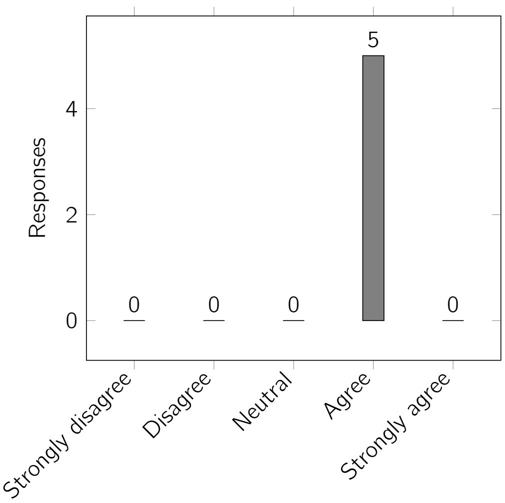
By analysing the responses, the perspective of the users about the selection of appropriate facility location quicker was strongly positive because they agree with importance and complexity of the decision (c.f. figure 6.1). When asked about the response time of the system, they expected better response time (c.f. figure 6.2).

Understandability

The understandability measurement tries to capture the capability of the system to enable the user to understand the system, its features and conditions of use. Were defined one question about this subject:

1. The results are easy to understand (c.f. figure 6.3).

Figure 6.3: Understandability - The results are easy to understand



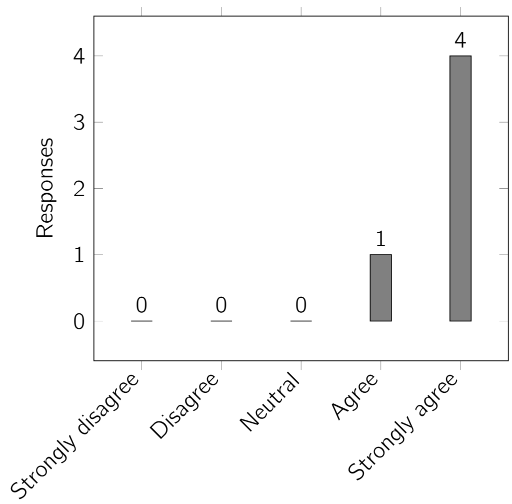
Analysing the responses, the perspective of the users about the ease of comprehension was positive (c.f. figure 6.3). In the future, this metric will be improved with the integration of the system with a web application that gives a graphical and organised perspective of the results.

Accuracy

The accuracy measurement tries to capture the correctness of the output information. Were defined four questions about this subject:

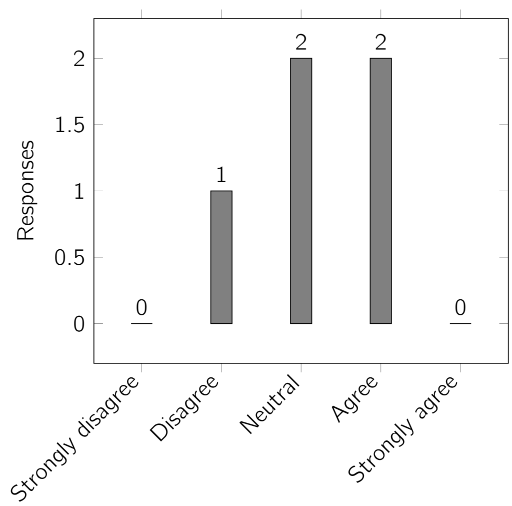
1. The facility location suggestions are reasonable (c.f. figure 6.4);

Figure 6.4: Accuracy - The facility location suggestions are reasonable



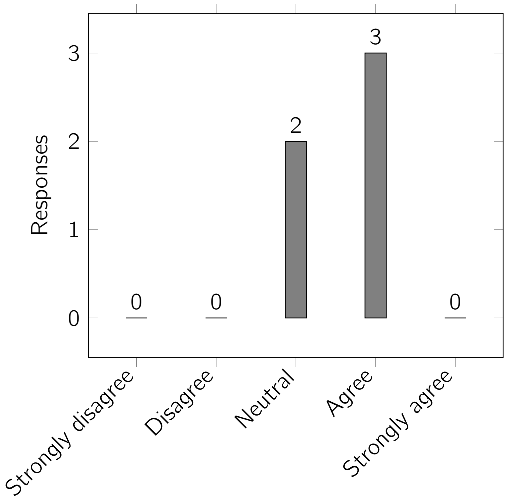
2. Parameters used to make the decision are sufficient (c.f. figure 6.5);

Figure 6.5: Accuracy - Parameters used to make the decision are sufficient



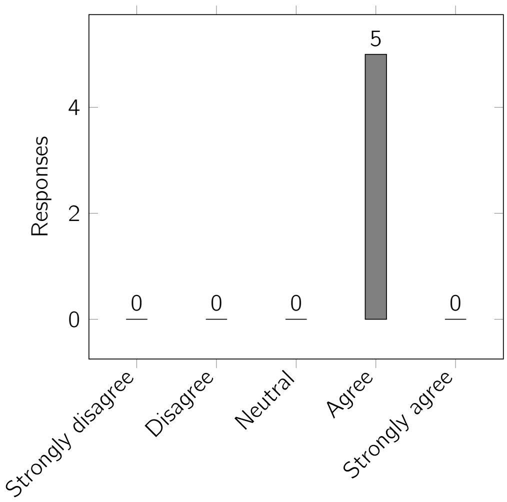
3. The facility location suggestions are accurate based on my business knowledge (c.f. figure 6.6);

Figure 6.6: Accuracy - The facility location suggestions are accurate based on my business knowledge



4. Information provided about the facility location suggestions is sufficient and clear (c.f. figure 6.7).

Figure 6.7: Accuracy - Information provided about the facility location suggestions is sufficient and clear



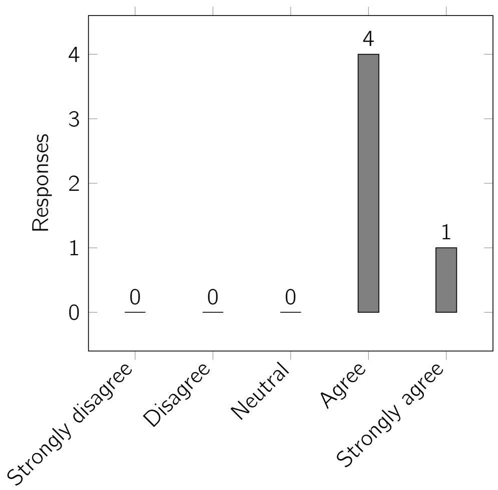
The perspective of the users about the facility location suggestions and accuracy of the suggestions based on business knowledge were positive because they visually agree with the provided suggestions since they can identify their most visited areas (c.f. figure 6.4 and figure 6.6). When asked about the sufficiency of the used parameters (c.f. figure 6.5) they consider that it would be interesting to have other factors such as average salaries and proximity to potential customers. Finally, they consider the information about the results of the facility location suggestions to be sufficient and explicit (c.f. figure 6.7).

Effectiveness

The effectiveness measurement tries to capture the capability of the system to achieve the specified goals. One question was defined about this subject:

1. Help in choosing appropriate facility location (c.f. figure 6.8).

Figure 6.8: Effectiveness - Help in choosing appropriate facility location



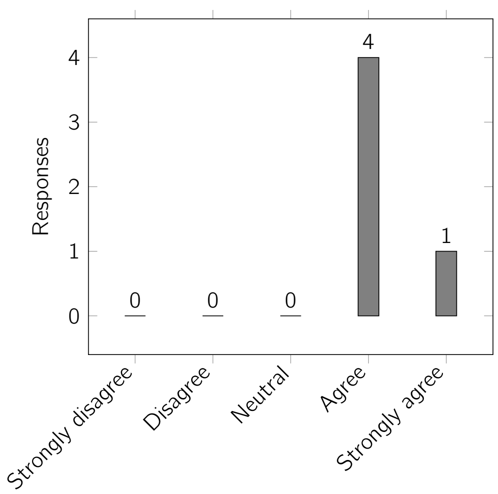
By analysing the responses, the perspective of the users about the assistance of the system by helping them choose appropriate facility location is favourable (c.f. figure 6.8).

Overall

The overall measurement tries to capture the global perception of the system. Were defined two questions about this subject:

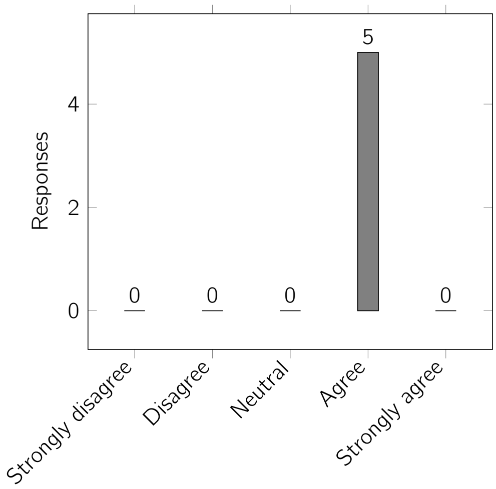
1. Achieved the goals to support the decision making on facility location selection (c.f. figure 6.9);

Figure 6.9: Overall Evaluation - Achieved the goals to support the decision making on facility location selection



2. Overall, I am satisfied with this system (c.f. figure 6.10).

Figure 6.10: Overall Evaluation - I am satisfied with this system



By analysing the responses, the perspective of the users about the overall functionalities is satisfactory (c.f. figure 6.9 and figure 6.10). They identify potential in this solution since they can use this system to support their facility location decisions and make use of persisted data transmitted by their vehicles.

6.4.2 Clustering Analysis

This section describes the approach to define the DBSCAN parameters by the execution and discussion of experiments and, lastly, discusses the Density-Based Clustering Validation.

Determining DBSCAN Parameters

Ideally, before development, we would have to know the values of *Eps* and *MinPts* for each cluster, but there is no easy way to get this information in advance for all clusters on the database (Ester et al. 1996).

Theoretically, a more considerable value for *Eps* results in broader clusters, while a smaller value produces smaller narrower clusters. Considering this solution wants to capture the most visited and dense areas, the *Eps* was defined to 1.5 kilometres.

Theoretically, a larger value of *MinPts* produces a bigger cluster but may exclude smaller areas as it attempts to merge them in one large cluster. This value wants to capture the most dense areas and there were made a few experiments (c.f. figures 6.4 and 6.5) to determine the *MinPts*.

These experiments were executed using the machine specifications according to figure 6.3 and the clustering algorithm was executed based on data sets provided by Company A and Company B. For each data set was considered the *MinPts* to be: 100; 150; 200; 2500.

Table 6.3: System Properties - Machine Specifications

RAM	Disk	Operating System
20GB	220GB SSD	Windows 10 Pro

The first scenario uses the data from Company A that has 160.542 data points gathered from the last six months. The data distribution of this data set can be found in figure 5.5.

Table 6.4: Company A - Experiments - MinPts definition

MinPts	Eps	Clusters	Execution (s)
100	1.5	90	58
150	1.5	74	58
200	1.5	60	57
2500	1.5	5	57

The second scenario uses the data from Company B, that has 140.533 data points gathered from the last six months. The data distribution of this data set can be found in figure 5.7.

Table 6.5: Company B - Experiments - MinPts definition

MinPts	Eps	Clusters	Execution (s)
100	1.5	88	93
150	1.5	60	72
200	1.5	45	75
2500	1.5	6	87

The executions for each chosen *MinPts* show that if the value increases, the number of clusters decrease since the higher the value, the more restricted is the condition to a potential cluster to be effectively considered a cluster. Contrariwise, more points are considered to be noise or outliers.

The last *MinPts* value - 2500 - is the one considered by domain to be the most suitable value according to the domain knowledge, since it has good results on the following expressions:

$$journeysPerMonth = \frac{MinPts}{Months}$$

$$journeysPerBusinessDays = \frac{journeysPerMonth}{Days}$$

So, according to the expressions, 2500 journeys divided by six months is approximately 416 journeys per month. The 416 journeys per month divided by 22 business days is approximately 19 journeys per day that are located maximum 1.5 km from the core point.

In summary, minimizing the number of relevant clusters by maximizing the minimum points surrounding a core point is crucial to simplify the interpretation and identify the results of the most relevant clusters. The chosen values for the DBSCAN parameters try to capture the densest areas by ensuring that a cluster has at least 2500 visits within 1.5 km radius on the last six months. The experiment results by executing the clustering algorithm with data sets from Company A and Company B, are on figures 5.6 and 5.8.

Density-Based Clustering Validation

Unfortunately, the Density-Based Clustering Validation was not performed due to computational limitations.

Table 6.6: System Properties - Machine Specifications

RAM	Disk	Operating System
20GB	220GB SSD	Windows 10 Pro
64GB DDR4	500GB SSD	Ubuntu 18

This method was executed using the machine specifications according to figure 6.6 but each execution resulted on out of memory error due to machine limitations possibly caused due to performance issues on the used algorithm that lead to this error or to data set size.

6.4.3 Software Testing

This section evaluates the Decision Support System solution through software testing by the description of the applied unit and load tests.

Unit Testing

The unit tests were developed for the Decision Support System by testing the decision service, the factor service, the location service and the suggestion service. A xUnit project was created, targeting the .NET Core 3.1 framework.

The table 6.8 shows the most relevant test scenarios that were developed focusing on the behaviour of the suggestion service. These tests simulates executions for three geospatial clusters and two factors (c.f. table 6.7) with varying weights.

Table 6.7: Suggestion Service - Unit test factors

Location	Cluster Size	Number of Crimes
X	250	71
Y	233	114
Z	300	90

The purpose is to test the TOPSIS calculation of the performance score for every three geospatial clusters.

Table 6.8: Suggestion Service - Unit test scenarios

Test	Weight 1	Weight 2	Performance Score
1	0.5	0.5	[0.25373134328358232; 0; 1]
2	0.3	0.7	[0.25373134328358238; 0; 1]
3	0.7	0.3	[0.25373134328358260; 0; 1]

Load Testing

There were made a few experiments to define the JMeter load testing parameters (c.f. table 6.9). Considering the current number of existing customers and their company sizes were simulated five users at the same time after 300 seconds of execution. The experiment executions were related to each available data set (Company A and Company B) and to all available factors (c.f. table 5.8).

Table 6.9: JMeter Load Testing configurations

Number of Threads	Ramp-Up Period (s)	Loop Count
5	300	1

The machine specifications used on the execution of these experiments is detailed on the table 6.10.

Table 6.10: System Properties - Machine Specifications

RAM	Disk	Operating System
20GB	220GB SSD	Windows 10 Pro

The results of the load test execution (c.f. table 6.11) shows that for Company A with 161.220 journeys, the average execution time was 65 seconds and for Company B with 140.535 journeys, the average execution time was 87 seconds. The total execution time is not exclusively linked to data size but also with data space distribution.

Table 6.11: Load Testing results - Time execution

Data set	Average (s)	Min (s)	Max (s)
Company A	68	65	70
Company B	87	85	90

The proof of concept developed for this thesis uses some libraries that are responsible for performing critical actions that require complex calculations leading to slow executions. The bottleneck problem of this implementation is the used DBSCAN algorithm provided by the

Scikit-learn that takes 80% to 90% of the total time execution that is not prepared to handle large data sets.

As described on the limitations (c.f. sections 7.2) and future work (c.f. section 7.3) sections, one of the most important steps towards a production solution is the development or use of a clustering algorithm implementation that is prepared to handle a significant volume of data. Finally, the system should use horizontal scaling to handle more workload.

6.4.4 Solution Experiment

This section details an execution scenario of generating facility location suggestions using the data from the Company A. This explanation is divided in three sections: (1) the parameters used to execute the program (c.f. section 6.4.4); (2) the software execution by detailing the processes to gather the alternatives, the criterion and the ordered suggestions (c.f. section 6.4.4); (3) the output results (c.f. section 6.4.4).

Input Parameters

To execute the mechanism responsible for generating the facility location suggestion it is necessary to call the entry point of the solution that is the suggestions controller from the Decision Support System API (c.f. table 5.1).

The method responsible to gather the facility location suggestions requires an encoded base64 query that holds the account identifier, the factor identifiers, the factor weights (the total sum of each weight must be equal to 1) and classification of the factors as cost or benefit (0 and 1 respectively).

The JSON structure at listing 6.1 represents an example of the structure that is converted to an encoded base64 query that is used to call the application.

```json
{
    "accountId": "companyA",
    "factorIds": [0, 1, 3, 5],
    "weights": [0.25, 0.25, 0.25, 0.25],
    "classifications": [1, 0, 0, 0]
}
```

Listing 6.1: Decision Support System - HTTP JSON Input of /suggestions

To simplify the comprehension of the used input parameters, the factors and their identifiers, the weights and the classifications are described on the table 6.12. This example uses four factors of the seven available on the system but the decision-maker can choose those who want to.

Table 6.12: Facility location factors - Identifiers, Weights and Classifications

Identifier	Factor	Weight	Classification
0	cluster-size	0.25	Benefit (1)
1	prices-per-sqf	0.25	Cost (0)
3	crime	0.25	Cost (0)
5	council-tax	0.25	Cost (0)

Software Execution

To produce the desired output it is necessary to gather alternatives and criterion to be used by the TOPSIS algorithm.

The alternatives are collected by analysing the available geospatial data from specified account. In this case, it is used data from the company A that has 161.220 journeys and 8.168.429 positions from the past six months. The table 6.13 describes the results from the clustering analysis performed by the Geospatial Analysis System API.

Table 6.13: Geospatial Analysis System API - Company A - Clusters

Latitude	Longitude	City	Post Code	Cluster Size
51.525507	-0.089147	London	EC1V 9FR	19696
51.741287	0.503573	Chelmsford	CM2 6UW	15516
51.522881	-0.44681	Uxbridge	UB8 3HE	7496
52.7158133	-2.4654983	Telford	TF1 7ES	6026

By visually analysing the data distribution on the figure 5.5 the data points are centred on London, and the clustering algorithm result (c.f. figure 5.6) shows that were found the three most significant clusters on that area.

Second, the criterion are collected based on the chosen factors by the decision-maker. The complete list of supported factors that can be used as the criterion on the MCDA is on the table 5.8 and the system responsible to gather the data for each selected factor is the Facility Location Factors System API (c.f. section 5.5). The values of the chosen factors for each cluster are detailed on the table 6.14.

Table 6.14: Facility Location Factors System API - Company A - Criterion

Post Code	Price Per Sqf	Crime	Council Tax
EC1V 9FR	1033	646	779.44
CM2 6UW	362	187	846.96
UB8 3HE	463	207	759.76
TF1 7ES	182	178	564.21

Finally, according to the MCDA, the used algorithm is the TOPSIS that is responsible to calculate the performance scores for the gathered alternatives and criterion. The results are presented on the table 6.15.

Table 6.15: Decision Support System API - Company A - MCDA Results

Post Code	Performance Score
EC1V 9FR	0.34545808711487636
CM2 6UW	0.76065077548997218
UB8 3HE	0.59360089845622255
TF1 7ES	0.65724064259285919

The table 6.15 shows that the location with the post code CM2 6UW has the better performance score (0.76065077548997218) according to the chosen factors.

Output Result

The full HTTP JSON response of the mechanism responsible to produce ordered facility location suggestions is on the listing 6.2 that follows the structure described on listing 5.1.

The response is an array of facility location suggestions that each object has the information about the identified cluster, the chosen factors and its values, and the performance score calculated by the TOPSIS algorithm.

```
[
    {
        "id": "b6e7fe11-64e2-41c5-ab0c-51551c7096dc",
        "timestamp": "2020-06-17T00:19:52.4524846Z",
        "location": {
            "postCode": "EC1V 9FR",
            "outCode": "EC1V",
            "inCode": "9FR",
            "latitude": 51.525507,
            "longitude": -0.089147,
            "clusterSize": 19696
        },
        "factors": [
            "clusterSize",
            "Price Per Sqf",
            "Crime",
            "Council Tax Band"
        ],
        "factorValuesByLocation": [
            19696.0,
            1033.0,
            646.0,
            779.44
        ],
        "performanceScore": 0.34545808711487636
    },
    {
        "id": "357db55a-2fb7-40f2-a9e0-c16ffd94e431",
        "timestamp": "2020-06-17T00:19:52.4524957Z",
        "location": {
            "postCode": "CM2 6UW",
            "outCode": "CM2",
            "inCode": "6UW",
            "latitude": 51.741287,
```

```
            "longitude": 0.503573,
            "clusterSize": 15516
        },
        "factors": [
            "clusterSize",
            "Price Per Sqf",
            "Crime",
            "Council Tax Band"
        ],
        "factorValuesByLocation": [
            15516.0,
            362.0,
            187.0,
            846.96
        ],
        "performanceScore": 0.76065077548997218
    },
    {
        "id": "78a3aaf7-a00d-40de-b067-75427316c846",
        "timestamp": "2020-06-17T00:19:52.452498Z",
        "location": {
            "postCode": "UB8 3HE",
            "outCode": "UB8",
            "inCode": "3HE",
            "latitude": 51.522881,
            "longitude": -0.44681,
            "clusterSize": 7496
        },
        "factors": [
            "clusterSize",
            "Price Per Sqf",
            "Crime",
            "Council Tax Band"
        ],
        "factorValuesByLocation": [
            7496.0,
            463.0,
            207.0,
            759.76
        ],
        "performanceScore": 0.59360089845622255
    },
    {
        "id": "99f5ce11-8ba6-4e4b-8b4f-dd71856088d5",
        "timestamp": "2020-06-17T00:19:52.4525001Z",
        "location": {
            "postCode": "TF1 7ES",
            "outCode": "TF1",
            "inCode": "7ES",
            "latitude": 52.7158133,
            "longitude": -2.4654983,
            "clusterSize": 6026
        },
        "factors": [
            "clusterSize",
            "Price Per Sqf",
            "Crime",
            "Council Tax Band"
        ],
```

```
        "factorValuesByLocation": [
            6026.0,
            182.0,
            178.0,
            564.21
        ],
        "performanceScore": 0.65724064259285919
    }
]
```


Listing 6.2: Decision Support System - HTTP JSON Response of /suggestions

6.5 Summary

To verify and validate the effectiveness of a DSS, it is fundamental to develop a system evaluation in order to verify that the system is capable of suggesting strategic, accurate and valuable facility locations based on fleet analysis and business needs.

The evaluation allows the developers to understand how well the system works and how it addresses the need of its users (Shafinah et al. 2010). The evaluation of the hypothesis focused on the quality of the DSS effectiveness results, clustering analysis and system load test analysis.

The Decision Support System quality analysis is evaluated through a survey that shows that the perspective of the users about the assistance of the system by helping them choose appropriate facility location is favourable. As stated on hypothesis, this analysis showed that the users agree about the accuracy and the value of the facility location suggestions.

The clustering analysis tries to capture the importance of the clustering algorithm parameters and the validation of the produced results. Setting these parameters correctly and verify the algorithm improves the accuracy of the suggested locations leading to strategic and valuable facility locations according to customer data.

To evaluate the clustering results was decided to determine the homogeneity, completeness, V-measure, Adjusted Rand Index and Adjusted Mutual Information, but these measurements require a parameter with the expected output to compare with the algorithm execution and since this is an unsupervised learning there is no correct solution to be compared with (c.f. section 6.3.2). The algorithm that could be used to evaluate the clustering were the DBCV but was not performed due to computational limitations (c.f. section 6.4.2).

The software testing determines the quality of the software by improving security, product quality and customer satisfaction. Testing procedures reduces maintenance costs, and the results are reliable and accurate.

Finally, in order to detail the procedure of the generation of facility location suggestions, the section software experiment describes the three main stages of the procedure: (1) input parameters; (2) software execution; and (3) output result.

Chapter 7

Conclusion

This chapter concerns the function of concluding this dissertation by giving an overview of the developed work, detailing the achieved requirements, discussing the limitations, enumerating and describing the future work and a final appreciation.

Fleet management is the process that fleet managers use to handle all fleet and asset data. Getting the best from a fleet and improving operational and financial performance is imperative. Data is one of the most important aspects of any fleet business. It helps to drive companies forward, keep the business flowing and optimise efficiencies.

As stated, the purpose of this project is to increase the value of Fleet Management solutions by adding an intelligent mechanism that can help companies to make better decisions based on the telemetry coming from their assets.

When a company is interested in creating or relocating a facility location, need to be aware of their business scope and all the related restrictions. It is imperative to analyse their client locations, journeys and be aware of the factors that may affect their geographical decision.

Facility location-allocation is a critical element and plays a role in the strategic design of business company's network (Melo, Nickel, and Saldanha-da-Gama 2009). Decisions related to the selection of the facility location are a critical element in strategic planning for many companies and often challenged by difficult spatial resource allocation decisions.

The developed solution is composed by three main components:

- Decision Support System that is responsible for producing facility location suggestions through the implementation of the multi-criteria decision analysis based on the decision-maker request.
- Geospatial Analysis System that is responsible to process telemetry data and to identify geospatial clusters. This data is provided to the Decision Support System to be used as alternatives in the multi-criteria decision analysis;
- Facility Location Factors System that is responsible for getting data from government data sources to each identified cluster. This data is provided to the Decision Support System to be used as criterion in the multi-criteria decision analysis.

This system retrieves suggestions of the most suitable facility locations according to the company's data. The decision-makers must select sites that will not merely perform well according to the current system state, but that will continue to be profitable for the facility's lifetime, even as environmental factors change, populations shift, and market trends evolve (Owen and Daskin 1998).

Analysing all these complex constraints can be often difficult without the support of any intelligent mechanism capable of providing suggestions to support their decisions. The decision-maker still has the responsibility to decide which is the best facility location since the system is only responsible for retrieving a set of suggestions that still require a final decision.

Some advantages of this solution are:

1. Supports the decision-maker to decide the most suitable facility location;
2. Automates and simplifies the complexity of the facility location decision;
3. Produces value by analysing existing data that is not being used;
4. Capable to ingest data from any source since it respects the defined data structure;
5. Capable of implementing factors from any source.

In summary, as demonstrated in chapter 5 and corroborated on chapter 6, the results show that the system is capable of suggestions strategic and valuable facility locations based on fleet analysis and business needs.

The output helps the business managers to make better decisions by returning locations that have potential to maximise the company's profit by reducing transportation and fuel costs and maximise the number of covered customers by expanding their territorial coverage.

7.1 Achieved Requirements

The main goal of this project was the development of a system capable of producing ordered facility location suggestions based on journeys made by a company and on chosen factors that the decision-maker considers more relevant on the selection of a facility location. Furthermore, other goals were the development of a solution that follows good software development practices by ensuring a system with quality, accuracy and effectiveness.

Requirements identification, analysis and design, were made before the construction and testing of the solution. This system follows good software development practises by using iterative and incremental development, managing dynamic requirements, component architecture to improve reuse, maintainability and extensibility. Finally, the quality was verified by testing the solution and by using a version control system.

The following objectives were accomplished according to the main objectives defined on section 1.3:

1. Study the problem's state of the art;
   (a) Facility location problems;
   (b) Machine Learning - focusing on unsupervised learning;
   (c) Decision Support System - focusing on multi-criteria decision analysis (MCDA) techniques;
   (d) Technologies - focusing on clustering algorithms;
2. Design and conception of a Decision Support System;

   (a) Implement a system of geospatial clustering analysis capable of identifying potential facility locations;
   (b) Study the factors that can affect the long-term facility location decision. Implement the identified and most relevant factors to be used by the DSS.
   (c) Implement a MCDA technique to produce ordered facility location suggestions.
3. Construction and development of a robust Decision Support System;
4. System integration with real telemetry data being transmitted by the assets;
5. Evaluate and test the solution. The evaluation focus on the quality analysis and performance of the DSS, and the metrics of the clustering algorithm.

This solution follows SOLID principles that were used to guide software development and even the architecture of the solution. According to the functional requirements defined in section 4.1.2 were accomplished the following objectives:

1. Use Case 1.1: The decision-maker initiates the decision support system mechanism;
2. Use Case 1.2: The decision-maker chooses the facility location;
3. Use Case 1.3: The decision-maker lists the facility location decisions.

According to the non-functional requirements defined in section 4.1.3 were accomplished almost all quality factors. The usability was not accomplished since this system requires future integration with a web application that was not the purpose of this project. The physical constraints were not satisfied since the solution is a prototype and was not deployed.

Finally, the performance needs to be improved by rethinking the implementation or the library used to perform the DBSCAN algorithm. Naturally, one of the goals is to speed up the system response time that is not uniquely correlated with the algorithm performance but also with high data volume that can lead to higher response times.

In summary, the overall objectives of this thesis were successfully accomplished.

7.2 Limitations

The developed system provides facility location suggestions but does not replace the decision-maker final decision. The decision-maker needs to be aware that the system only analyses the gathered data based on the request made by the user and may not be conscious of the real-world constraints and other non-requested factors.

Another limitation of the developed system is that the clustering implementation uses some libraries that are responsible for performing critical actions that require complex calculations leading to slow executions. The bottleneck problem of this implementation is the used DBSCAN algorithm provided by the Scikit-learn that takes 80% to 90% of the total time execution that is not prepared to handle large data sets (c.f. section 6.4.3).

Finally, each company has its geographical data distribution making each unique execution leading to different data processing and outputs. Since the data can have various distributions, shapes and density levels, the chosen parameters on DBSCAN could not be accurate to some execution scenarios. To ensure the parameters were correctly chosen, more data from other companies need to be tested. For this project, these parameters were tuned based on data provided by two companies.

7.3 Future Work

In order to improve continuously the develop work, the following items are considered important to allow the solution to reach more stability in order to be ready to be launched to be market:

- Improve the calculation of the TAM, SAM and SOM on the value analysis. Due to lack of available data it was not possible to make a correlation between the number of employees, the business industries and the number of assets per business;
- Add proximity to potential customers to the available list of factors;
- Add rate to the quality of the roads to the available list of factors;
- Add fuel costs to the available list of factors;
- Improve system speed and performance focusing on the DBSCAN clustering algorithm. The first approach to solve this issue should be using an existing implementation of the algorithm using other frameworks. In case of unsatisfactory results, other clustering algorithms or approach should be considered;
- Add support for new countries. Currently, this project supports factors for the United Kingdom and is sufficient for the Fonix Telematics market strategy. This task will only be necessary in case of market and geographical expansion;
- Improve the Decision Support System quality analysis by increasing the number of respondents (Faulkner 2003).

This system will need be integrated with a web application and in order to improve the user experience, the mechanism responsible to produce facility location suggestions can be made as a background task. In the future, one possible integration with the web application is:

1. The user triggers the suggestions mechanism by sending a message to a queue;
2. The DSS system is listening to a queue and triggers the internal mechanism;
3. When finished, the results are sent to the user through notification.

This possibility gives the users the capability to trigger a background mechanism without blocking their actions on the application they are using. In summary, future work needs to be done in order to prepare the solution to the production environment and add value to the current Fonix Telematics fleet management solution.

7.4 Final Appreciation

This project was challenging, and its complexity kept the student motivated from the beginning since it was required to study new fields of Computer Science. The most challenging branches were the unsupervised learning focusing on the clustering algorithms, clustering techniques, data analysis and new technologies.

Regarding the supervisors' guidance, the work was planned correctly along with the professors. Meetings were arranged to guide the student through the process. Over the development of the thesis, the supervisors were always helpful by making suggestions and corrections to the developed work.

Relatively to the developed solution, getting the best from a fleet and improving operational and financial performance is imperative. Data is one of the most important aspects of any fleet business. It helps to drive companies forward, keep the business flowing and optimise efficiencies. This system interprets and acts on data helping the clients to make better decisions.

According to the achieved results, the output helps the business managers to make better decisions by returning locations that have potential to maximise the company's profit by reducing transportation and fuel costs and maximise the number of covered customers by expanding their territorial coverage.

Finally, it was gratifying to solve problems with this complexity that have the potential to create and add value to the current organisation.

Bibliography

Aliyeva, Kamala (2018). "FACILITY LOCATION PROBLEM BY USING FUZZY TOPSIS METHOD". In: *Chemical Technology. Control and Management* 2018.3, pp. 55–59. url: https://uzjournals.edu.uz/ijctcmAvailableat:https://uzjournals.edu.uz/ijctcm/vol2018/iss3/13.

Ankerst, Mihael et al. (1999). *OPTICS: Ordering Points to Identify the Clustering Structure*. Tech. rep. 2, pp. 49–60. doi: 10.1145/304181.304187.

Aruldoss, Martin, T Miranda Lakshmi, and V Prasanna Venkatesan (2013). "A Survey on Multi Criteria Decision Making Methods and Its Applications". In: *American Journal of Information Systems* 1.1, pp. 31–43. doi: 10.12691/ajis-1-1-5. url: http://pubs.sciepub.com/ajis/1/1/5.

Babur, Iftikhar Hussain et al. (2015). "Analysis of Dbscan Clustering Technique on Different Datasets Using Weka Tool". In: *Sci.Int.(Lahore)* 27.6, pp. 5087–5090.

Berkhin, Pavel (2013). *Survey_of_Clustering_Data_Mining_Techniques.pdf*. Tech. rep., p. 56.

Boeing, Geoff (2018). "Clustering to Reduce Spatial Data Set Size". In: *SSRN Electronic Journal* March, pp. 1–7. issn: 1556-5068. doi: 10.2139/ssrn.3145515. arXiv: 1803.08101.

Business Model Canvas (2020). url: https://upload.wikimedia.org/wikipedia/commons/1/10/Business_Model_Canvas.png (visited on 01/23/2020).

Cunningham, Pádraig, Matthieu Cord, and Sarah Jane Delany (2008). *Supervised learning*. Tech. rep., pp. 21–49. doi: 10.4135/9781483399782.n6.

Deployment Diagram (2020). url: https://creately.com/blog/diagrams/deployment-diagram-tutorial/ (visited on 02/20/2020).

Dönmez, Pınar (2013). *Introduction to Machine Learning*. Tech. rep. 2. doi: 10.1017/s1351324912000290.

Dumas, Joseph and Janice Redish (1993). *A Practical Guide to Usability Testing*.

Eeles, Peter (2004). *What , no supplementary specification ?* url: https://www.ibm.com/developerworks/rational/library/3975.html (visited on 02/19/2020).

Ester, Martin et al. (1996). *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. Tech. rep. url: www.aaai.org.

Fabris, Fabio, João Pedro de Magalhães, and Alex A. Freitas (2017). "A review of supervised machine learning applied to ageing research". In: *Biogerontology* 18.2, pp. 171–188. issn: 15736768. doi: 10.1007/s10522-017-9683-y.

Farahani, Reza Zanjirani and Masoud Hekmatfar (2009). *Dynamic facility location problem*. isbn: 978-3-7908-2150-5. doi: 10.1007/978-3-7908-2151-2_15. url: https://www.researchgate.net/publication/227149037.

Faulkner, Laura (2003). *Beyond the five-user assumption: Benefits of increased sample sizes in usability testing*. Tech. rep. 3. Austin, Texas: University of Texas, pp. 379–383. doi: 10.3758/BF03195514.

Felsberger, Andreas, Bernhard Oberegger, and Gerald Reiner (2017). “A review of decision support systems for manufacturing systems”. In: *CEUR Workshop Proceedings* 1793.February 2017. issn: 16130073.

Fleet Management Solutions Market | Growth, Trends, and Forecast (2020-2025) (2019). url: https://www.mordorintelligence.com/industry-reports/global-fleet-management-software-market-industry (visited on 01/19/2020).

Frazão, Talita D.C. et al. (2018). “Multicriteria decision analysis (MCDA) in health care: A systematic review of the main characteristics and methodological steps”. In: *BMC Medical Informatics and Decision Making* 18.1. issn: 14726947. doi: 10.1186/s12911-018-0663-1. url: https://doi.org/10.1186/s12911-018-0663-1.

Gupta, Alind (2020). *ML | OPTICS Clustering Explanation*. url: https://www.geeksforgeeks.org/ml-optics-clustering-explanation/ (visited on 02/13/2020).

Halkidi, Maria, Yannis Batistakis, and Michalis Vazirgiannis (2001). “On Clustering Validation Techniques”. In: *Journal of Intelligent Information Systems* 17.3, pp. 107–145. issn: 00138703. doi: 10.1023/A.

Hosack, Bryan et al. (2012). “A look toward the future: Decision support systems research is alive and well”. In: *Journal of the Association for Information Systems* 13.5, pp. 315–340. issn: 15369323. doi: 10.17705/1jais.00297.

Hwang, Ching-Lai and Kwangsun Yoon (1981). “Methods for Multiple Attribute Decision Making”. In: *Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey*. Berlin, Heidelberg: Springer Berlin Heidelberg. isbn: 978-3-642-48318-9. doi: 10.1007/978-3-642-48318-9_3. url: https://doi.org/10.1007/978-3-642-48318-9%7B%5C_%7D3.

Jain, Anil K. and Richard C. Dubes (1988). *Algorithms for Clustering Data*. Ed. by Barbara Marttine, Karen Stephens, and Lorraine Fumoso. Michigan: Prentice-Hall, Inc., p. 320. isbn: 0-13-022278-X. url: https://homepages.inf.ed.ac.uk/rbf/BOOKS/JAIN/Clustering%7B%5C_%7DJain%7B%5C_%7DDubes.pdf.

Jensen, Anker (2012). *VALUE CONCEPTS AND VALUE BASED COLLABORATION IN BUILDING PROJECTS*. Tech. rep. Technical University of Denmark, Denmark.

Jozaghi, Ali et al. (2018). “A comparative study of the AHP and TOPSIS techniques for dam site selection using GIS: A case study of Sistan and Baluchestan Province, Iran”. In: *Geosciences (Switzerland)* 8.12. issn: 20763263. doi: 10.3390/geosciences8120494. url: www.preprints.org.

Kaliszewski, I and D Podkopaev (2016). “Simple additive weighting—A metamodel for multiple criteria decision analysis methods”. In: *Expert Systems with Applications* 54, pp. 155–161. issn: 0957-4174. doi: https://doi.org/10.1016/j.eswa.2016.01.042. url: http://www.sciencedirect.com/science/article/pii/S0957417416000658.

Kant, Krishna et al. (2018). “A Two Step Clustering Method for Facility Location Problem”. In: *International Journal of Advanced Intelligence Paradigms* 10.1, p. 1. issn: 1755-0386. doi: 10.1504/ijaip.2018.10011478.

Koen, Peter A, Greg M Ajamian, et al. (2001). *FuzzyFrontEnd: Effective Methods, Tools, and Techniques LITERATURE REVIEW AND RATIONALE FOR DEVELOPING THE NCD MODEL*. Tech. rep. url: http://www.stevens-tech.edu/cce/NEW/PDFs/FuzzyFrontEnd%7B%5C_%7DOld.pdf.

Koen, Peter A, Heidi M J Bertels, and Elko J Kleinschmidt (2014). “Managing the Front End of Innovation”. In: pp. 25–35. doi: 10.5437/08956308X5703199. url: https://pdfs.semanticscholar.org/cbc5/55272cf739ffe6cb379fd5c8d545121c721a.pdf.

Koen, Peter et al. (2001). *PROVIDING CLARITY AND A COMMON LANGUAGE TO THE "FUZZY FRONT END"*. Tech. rep.

Kotler, P and K Keller (2006). *Marketing management*.
Kumar, Mukesh and Amandeep Verma (2018). "Text Clustering Techniques A Review". In: *International Journal of Computer Sciences and Engineering* 6.6, pp. 1091–1099. doi: 10.26438/ijcse/v6i6.10911099.
Lavrakas, Paul (Apr. 2008). "Research Hypothesis". In: *Encyclopedia of Survey Research Methods*. Sage Publications, Inc. doi: 10.4135/9781412963947.n472.
Mareschal, Bertrand et al. (2005). *PROMETHEE methods*. Vol. 78. May 2014, pp. 163–195. isbn: 0387230815. doi: 10.1007/0-387-23081-5_5.
Mayank, Mohit (Mar. 2018). *Reinforcement Learning with Q tables by Mohit Mayank*. url: https://itnext.io/reinforcement-learning-with-q-tables-5f11168862c8 (visited on 09/30/2020).
MCDA (2011). url: https://projects.ncsu.edu/nrli/decision-making/MCDA.php (visited on 02/04/2020).
Melo, Teresa, Stefan Nickel, and Francisco Saldanha-da-Gama (July 2009). "Facility Location and Supply Chain Management – a Review". In: *European Journal of Operational Research* 196, pp. 401–412. doi: 10.1016/j.ejor.2008.05.007.
MIT (2012). "Calculate the Total Addressable Market (TAM) Size for the Beachhead Market". In: *Innovation-Based Entrepreneurship 101: A Handbook on 24 Steps to Help Make Entrepreneurs More Successful*, pp. 44–51.
Mocenni, Chiara (2011). *The Analytic Hierarchy Process*. Tech. rep.
MongoDB (2020). url: https://www.mongodb.com/what-is-mongodb (visited on 02/20/2020).
Moulavi, Davoud et al. (2014). *Density-Based Clustering Validation*. Tech. rep. Philadelphia, pp. 1–9.
Nguyen, Giang et al. (2019). "Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey". In: *Artificial Intelligence Review* 52, pp. 77–124. doi: 10.1007/s10462-018-09679-z. url: https://doi.org/10.1007/s10462-018-09679-z.
Nilsson, Nils J (1998). *Introduction to Machine Learning: An Early Draft of a Proposed Textbook.* Tech. rep., p. 179. url: http://ai.stanford.edu/people/nilsson/MLBOOK.pdf.
Owen, Susan Hesse and Mark S. Daskin (1998). *Strategic facility location: A review*. Tech. rep. 3, pp. 423–447. doi: 10.1016/S0377-2217(98)00186-6.
Panda, Monalisa and Alok Kumar Jagadev (2018). "TOPSIS in Multi-Criteria Decision Making: A Survey". In: *Proceedings - 2nd International Conference on Data Science and Business Analytics, ICDSBA 2018*, pp. 51–54. doi: 10.1109/ICDSBA.2018.00017.
PostgreSQL (2020). url: https://www.postgresql.org/ (visited on 07/15/2020).
PyTorch (2020). url: https://pytorch.org/ (visited on 02/11/2020).
QFD Online (2019). url: http://www.qfdonline.com/templates/3f2504e0-4f89-11d3-9a0c-0305e82c2899/ (visited on 02/13/2020).
Rantell, Anthony (2019). *Fonix Telematics*. url: https://www.fonixtelematics.com/.
Saaty, R. W. (1987). "The analytic hierarchy process-what it is and how it is used". In: *Mathematical Modelling* 9.3-5, pp. 161–176. issn: 02700255. doi: 10.1016/0270-0255(87)90473-8.
Salkind, Neil (2013). "Cluster Analysis". In: *Encyclopedia of Measurement and Statistics*. doi: 10.4135/9781412952644.n87.
Saxena, Amit et al. (2017). "A review of clustering techniques and developments". In: *Neurocomputing* 267.July, pp. 664–681. issn: 18728286. doi: 10.1016/j.neucom.2017.06.053. url: http://dx.doi.org/10.1016/j.neucom.2017.06.053.

scikit-learn (2020). url: https://scikit-learn.org/stable/ (visited on 02/11/2020).

Scikit-Learn Haversine (2020). url: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.haversine%7B%5C_%7Ddistances.html?highlight=haversine%7B%5C#%7Dsklearn.metrics.pairwise.haversine%7B%5C_%7Ddistances (visited on 08/18/2020).

Shafinah, K et al. (2010). "System Evaluation for a Decision Support System". In: *Information Technology Journal* 9.5, p. 11. doi: 10.1007/978-3-540-48716-6.

Shatnawi, Ali et al. (2018). "A comparative study of open source deep learning frameworks". In: *2018 9th International Conference on Information and Communication Systems, ICICS 2018* 2018-January, pp. 72–77. doi: 10.1109/IACS.2018.8355444. arXiv: 1903.00102. url: https://www.researchgate.net/publication/324454774.

Shogun Machine Learning (2020). url: https://www.shogun.ml/ (visited on 02/11/2020).

Shukla, Om Ji, Lalit Upadhayay, and Amit Dhamija (2019). "Multi Criteria Decision Analysis Using AHP Technique to Improve Quality in Service Industry: An Empirical Study". In: 2019.March 2015. doi: 10.13140/RG.2.1.2747.0881.

Simon, Annina et al. (2015). "An Overview of Machine Learning and Its Applications". In: *International Journal of Electrical Sciences & Engineering (IJESE)* 1.1, pp. 22–24. doi: 10.1007/978-3-658-23751-6_17.

Sutton, Richard S and Andrew G Barto (2015). *Reinforcement Learning: An Introduction*, pp. 1–352.

Taherdoost, Hamed (2018). *Decision Making Using the Analytic Hierarchy Process (AHP); A Step by Step Approach*. Tech. rep. January 2017, pp. 244–246. url: https://www.researchgate.net/publication/322887394.

TensorFlow (2020). url: https://www.tensorflow.org/ (visited on 02/11/2020).

Twin, Alexandra (2019). *Value Proposition Definition*. url: https://www.investopedia.com/terms/v/valueproposition.asp (visited on 01/21/2020).

Ulaga, Woflgang and Andreas Eggert (2006). "Relationship value and relationship quality". In: *European Journal of Marketing* 40, pp. 311–327. doi: 10.1108/03090560610648075. url: http://dx.doi.org/10.1108/03090569610106626http://dx.doi.org/10.1108/03090560710752429.

W., J., George R. Davies, and Dale Yoder (2019). "Business Statistics". In: *The Mathematical Gazette* 22.251, p. 413. issn: 00255572. doi: 10.2307/3607925.

Warwick Manufacturing, Group (2007). *Quality Function Deployment*, pp. 1–62.

Widianta, M. M.D. et al. (2018). "Comparison of Multi-Criteria Decision Support Methods (AHP, TOPSIS, SAW & PROMENTHEE) for Employee Placement". In: *Journal of Physics: Conference Series* 953.1, p. 6. issn: 17426596. doi: 10.1088/1742-6596/953/1/012116.

Appendix A

Evaluation - Decision Support System Survey

Instructions: Please select (X) the option that describes your perception about the Decision Support System (DSS) for facility location problems in fleet management.

1. **Efficiency**

   (a) Help to choose appropriate facility location more quickly.

   O (1) Strongly disagree

   O (2) Disagree

   O (3) Neither agree nor disagree

   O (4) Agree

   O (5) Strongly agree

   (b) Response time of the system is acceptable.

   O (1) Strongly disagree

   O (2) Disagree

   O (3) Neither agree nor disagree

   O (4) Agree

   O (5) Strongly agree

2. **Understandability**

   (a) The results are easy to understand.

   O (1) Strongly disagree

   O (2) Disagree

   O (3) Neither agree nor disagree

   O (4) Agree

   O (5) Strongly agree

3. **Accuracy**
   (a) The facility location suggestions are reasonable.
      O (1) Strongly disagree
      O (2) Disagree
      O (3) Neither agree nor disagree
      O (4) Agree
      O (5) Strongly agree
   (b) Parameters used to make the decision are sufficient.
      O (1) Strongly disagree
      O (2) Disagree
      O (3) Neither agree nor disagree
      O (4) Agree
      O (5) Strongly agree
   (c) The facility location suggestions are accurate based on my business knowledge.
      O (1) Strongly disagree
      O (2) Disagree
      O (3) Neither agree nor disagree
      O (4) Agree
      O (5) Strongly agree
   (d) Information provided about the facility location suggestions is sufficient and clear.
      O (1) Strongly disagree
      O (2) Disagree
      O (3) Neither agree nor disagree
      O (4) Agree
      O (5) Strongly agree

4. **Effectiveness**
   (a) Help in choosing appropriate facility location.
       O (1) Strongly disagree
       O (2) Disagree
       O (3) Neither agree nor disagree
       O (4) Agree
       O (5) Strongly agree
5. **Overall Evaluation**
   (a) Achieved the goals to support the decision making on facility location selection.
       O (1) Strongly disagree
       O (2) Disagree
       O (3) Neither agree nor disagree
       O (4) Agree
       O (5) Strongly agree
   (b) Overall, I am satisfied with this system.
       O (1) Strongly disagree
       O (2) Disagree
       O (3) Neither agree nor disagree
       O (4) Agree
       O (5) Strongly agree